Università degli Studi di Salerno

Dottorato di Ricerca in Informatica e Ingegneria dell'Informazione
Ciclo 31 – a.a 2017/2018

TESI DI DOTTORATO / PH.D. THESIS

Model Checking Cyber-Physical Systems

YOUSSEF DRIOUICH

SUPERVISOR: **PROF. DOMENICO PARENTE**

PHD PROGRAM DIRECTOR: **PROF. PASQUALE CHIACCHIO**

Dipartimento di Ingegneria dell'Informazione ed Elettrica
e Matematica Applicata

Abstract

Cyber-Physical Systems (CPSs) are integrations of computation with physical processes. Applications of CPS arguably have the potential to overshadow the 20-th century IT revolution. Nowadays, CPSs application to many sectors like Smart Grids, Transportation, and Health help us run our lives and businesses smoothly, successfully and safely.
Since malfunctions in these CPSs can have serious, expensive, sometimes fatal consequences, Simulation-based Verification (SBV) tools are vital to minimize the probability of errors occurring during the development process and beyond. Their applicability is supported by the increasingly widespread use of Model Based Design (MBD) tools. MBD enables the simulation of CPS models in order to check for their correct behaviour from the very initial design phase. The disadvantage is that SBV for complex CPSs is an extremely resources and time-consuming process, which typically requires several months of simulation. Current SBV tools are aimed at accelerating the verification process with multiple simulators working simultaneously. To this end, they compute all the scenarios in advance in such a way as to split and simulate them in parallel. Nevertheless, there are still limitations that prevent a more widespread adoption of SBV tools. To this end, we present a MBD methodology aiming the acausual modeling and verification via formal-methods, specifically the model checking techniques, the system under verification (SUV). Our approach relies basically on: Firstly, the analysis of the steady-states of the CPS and the bounding technique of the system's state in parallel with the simulation in order to identify the state space of the

system simulating it only once, then represent it as a Finite State Machine (FSM). Secondly, exhaustively verify the resulted FSM using a symbolic model checker and express the desired properties in classical temporal logic. The application to a power management system is presented as a case study.

Sommario

I sistemi ciberfisici (CPS) sono integrazioni del calcolo con i processi fisici. Le applicazioni dei CPS probabilmente hanno il potere di sovrastare la rivoluzione IT del XX secolo. Oggigiorno, l'applicazione dei sistemi ciberfisici a molti settori come Smart Grid, trasporto e salute, ci aiuta a gestire le nostre vite e le nostre attività senza problemi, con successo e in sicurezza.
Poiché i malfunzionamenti in questi sistemi ciberfisici possono avere conseguenze gravi, costose e talvolta fatali, gli strumenti di verifica basata sulla simulazione (SBV) sono fondamentali per ridurre al minimo la probabilità di errori che si verificano durante il processo di sviluppo e oltre. La loro applicabilità è supportata dall'uso sempre più diffuso di strumenti Model Based Design (MBD). MBD consente la simulazione di modelli CPS al fine di verificare il loro corretto comportamento sin dalla fase di progettazione iniziale. Lo svantaggio è che la verifica basata sulla simulazione per CPS complessi è un processo estremamente dispendioso in termini di tempo e risorse, che in genere richiede diversi mesi di simulazione. Gli attuali strumenti SBV mirano ad accelerare il processo di verifica con più simulatori che lavorano simultaneamente. A tal fine, calcolano tutti gli scenari in anticipo in modo tale da suddividerli e simularli in parallelo. Tuttavia, esistono ancora limitazioni che impediscono un'adozione più diffusa degli strumenti di verifica basata sulla simulazione. A tal fine, presentiamo una metodologia MBD che mira alla modellazione e alla verifica acausual tramite metodi formali, in particolare le tecniche di model checking, il sistema sotto verifica. Il nostro approccio si basa essenzialmente su: In primo luogo, l'analisi degli

stati stazionari del CPS e la tecnica di delimitazione dello stato del sistema in parallelo con la simulazione al fine di identificare lo spazio dello stato del sistema simulandolo solo una volta, quindi rappresentarlo come una macchina a stati finiti(FSM). In secondo luogo, verificare esaurientemente l'FSM risultante utilizzando un symbolic model checker ed esprimere le proprietà desiderate nella logica temporale classica. L'applicazione a un power management system viene presentata come caso di studio.

Acknowledgements

Firstly, I would like to express my sincere gratitude to my advisor prof. Domenico Parente for the continuous support of my Ph.D study and related research, for his patience, motivation, and knowledge. His guidance helped me in all the time of research and writing of this thesis.

Besides my advisor, I would like to thank prof. Enrico Tronci for his insightful comments and encouragement, but also for the hard question which incented me to widen my research from various perspectives.

I would also like to thank the committee members, Dr. Toni Mancini, prof. Francesco Basile and prof. Gabriel Alfonso Rincón-Mora for serving as my committee members even at hardship.

A special thanks goes to Fabio, Angela, Cristina, Carmen and Giuseppe for their great help and assistance.

I would also like to thank my girlfriend Maria Rosaria for supporting me, I can't thank you enough for encouraging me throughout this experience.

Last but not the least, I would like to thank my family: my parents, my brother Ilias and my sister Imane for supporting me spiritually throughout writing this thesis and in my life in general, words can not express how grateful I am, they are the most important people in my world and I dedicate this thesis to them.

Contents

List of Figures

The seeker after truth is not one who studies the writings of the ancients and, following his natural disposition, puts his trust in them, but rather the one who suspects his faith in them and questions what he gathers from them, the one who submits to argument and demonstration and not the sayings of human beings whose nature is fraught with all kinds of imperfection and deficiency. Thus the duty of the man who investigates the writings of scientists, if learning the truth is his goal, is to make himself an enemy of all that he reads, and, applying his mind to the core and margins of its content, attack it from every side. He should also suspect himself as he performs his critical examination of it, so that he may avoid falling into either prejudice or leniency.

- Ibn al-Haytham -

2

Chapter 1

Introduction

In some key industries, such as defense, automobiles, medical devices, and the smart grid, the bulk of the innovations focus on cyber-physical systems. A key characteristic of cyber-physical systems is the close interaction of software components with physical processes, which impose stringent safety and time/space performance requirements on the systems[1] . Cyber-physical systems are safety-critical since violations of their requirements, such as missed deadlines or component failures, may have life-threatening consequences[2]. For example, when a cyber-physical system in a car detects a crash, the airbag must inflate in less than few milliseconds to avoid severe injuries to the driver. For this reason, verification activities are performed in parallel with the development process, from the initial design up to the acceptance testing stages.

1.1 Cyber-Physical Systems

A cyber-physical system (CPS) is an integration of computation with physical processes. Embedded computers and networks monitor and control the physical processes, usually with feedback loops where physical processes affect computations and vice versa. As an intellectual challenge, CPS is about the intersection, not the union, of the physical and the cyber. It is not sufficient to sepa-

rately understand the physical components and the computational components. We must instead understand their interaction. The design of this kind of systems is divided on three main parts: **modeling**, **design**, and **analysis**.
Modeling is the process of gaining a deeper understanding of a system through imitation. Models imitate the system and reflect properties of the system. Models specify what a system does. Design is the structured creation of artifacts. It specifies how a system does what it does. Analysis is the process of gaining a deeper understanding of a system through dissection. It specifies why a system does what it does (or fails to do what a model says it should do).

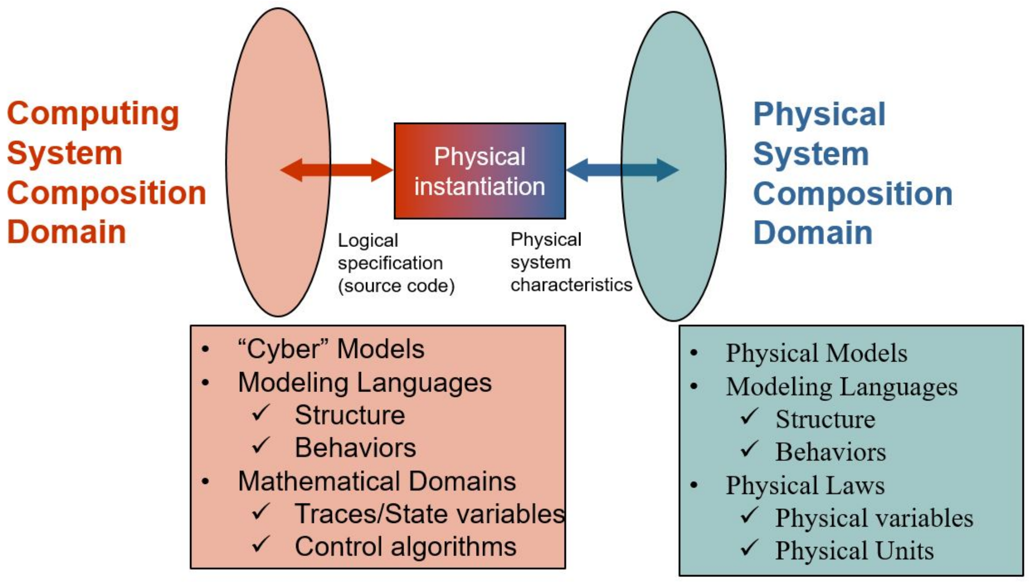


Figure 1.1 Cyber-Physical Systems representation

In the modeling process, we define a system to be a combination of parts that is considered a whole. A physical system is one realized in matter, in contrast to a conceptual or logical system such as software and algorithms. The dynamics of a system is its evolution in time: how its state changes. A model of a physical system is a description of certain aspects of the system that is intended to yield insight into properties of the system. Models have mathematical properties that enable systematic analysis. The model imitates properties of the system, and hence yields insight into that system. A model is itself a system. It is important to avoid confusing a model and the system that it models. These are two distinct artifacts. A model of a system is said to have high fidelity if it accurately describes properties of the system. It is said to abstract the system if it omits details. In this work, models of physical systems inevitably do omit details, so they are always abstractions of the system. Models of CPS normally include all three parts. The models will typically need to represent both dynamics and static properties (those that do not change during the operation of the system). Every system must be designed to meet certain requirements. For CPS, which are often intended for use in safety-critical, everyday applications, it is essential to certify that the system meets its requirements. Such system requirements are also called properties or specifications. It is important to precisely state requirements to avoid ambiguities inherent in natural languages. The goal of the many verification techniques is to help replace descriptive techniques with formal ones, which we believe are less error prone. Importantly, formal specifications also enable the use of automatic techniques for formal verification of both models and implementations.

1.2 Model Based Design

Model-based design (MBD) [3] emphasizes mathematical modeling to design, analyze, verify, and validate dynamic systems. A complete model of a cyber-physical system represents the coupling of its environment, physical processes, and embedded computations. Modeled systems may be tested and simulated offline, allowing developers to verify the logic of their application, assumptions about its environment, and end-to-end (i.e. closed-loop) behavior.
The design of a complex cyber-physical system is a challenging process. Commonly employed design techniques are worldly-wise

and include mathematical modeling of physical systems, formal models of computation, simulation of heterogeneous systems, software synthesis, verification, validation, and testing. In this work we have adopted a set of sequential steps that, if followed carefully and correctly, embraces each of these design techniques and sufficiently rules the development of a complex cyber-physical system.

1.2.1 System's Definition

Model Physical Processes: A first iteration of physical modeling should take in consideration basic observations and insight into pertinent physical systems, such as the environment in which the cyber-physical system operates, or physical processes to be controlled. Models of physical processes are simplified representations of real systems, and are usually in the form of systems of differential equations or continuous equations functions[4]. What may begin as simple mathematical models may need to be improved following development of a control algorithm, specification of hardware, and testing of components and subsystems.

Identify the Problem: Recluse fixed parameters, adjustable parameters, and controlled variables. Identify quantities that characterize physical processes, such as configuration spaces, safety limitations, input and output sets, saturation points, and modal behavior. Understand how a physical process may interact with a computation, for example system requirements, fault conditions, and reactions to noise and quantization.

Derive a Control Algorithm: Specify the conditions under which physical processes are controllable and determine a convenient control algorithm to be executed by an embedded computer. Use the problem characterization to specify requirements on sampling rates, delays, and quantization so that the desired physical dynamics can be precisely measured and perfectly controlled; these requirements must be satisfied by the computational platform used. In highly distributed applications, or systems that are generally asynchronous but locally synchronous, it may be mandatory to select models of computation before a control algorithm can be implemented. Reanalyze this step after selecting models of computation and specifying hardware to determine the impact of variable sampling estimations introduced by an asynchronous model of computation, or other nonlinear behaviours in the physical model.

1.2.2 Simulate, Validate and Verify

Simulate : Deal with the problem using a simulation platform. If multiple models of computation are to be used, simulation and synthesis tools must grant the compositions of and interactions between multiple models of computation. Depending on the robustness of the development environment, controllers, and physical processes. Use platform-based design to isolate application logic and architecture software into interchangeable components, which can improve code portability, decrease the impact of changing hardware components, and allow components to be reused in other applications. Models of individual components and subsystems are as crucial as a complete end-to-end model. Component models allow a test harness for construction, verification of synthesized software, and testing. If no modeling tool can totally describe the system, then for each subsystem use the modeling tool that best captures its dynamics. Countless simulation tools exist in literature, but most are bounded to only a minor models of computation and are unable to apprehend the interactions between heterogeneous systems.

Validate and Verify : Compose suitable parameters to establish test environments that are as straightforward as possible, and test each component and subsystem separately. Computational systems may be isolated from physical systems via hardware-in-the-loop measurement, where programmable hardware as instance embedded computers simulate the observation from physical or other computational processes. Measurements of execution time and latency can be used to tune earlier models, and unexpected test results may lead to erroneous modeling or implementation. Formal verification and validation give insight into the behavior of an algorithm over all or certain combinations of its inputs, or over the course of time. Precisely assert requirements and express them into a formal specification for verification and validation. List invariants that have to be verified during test phase. Verification and validation are likely the most complicated facets in the design of a cyber-physical system.

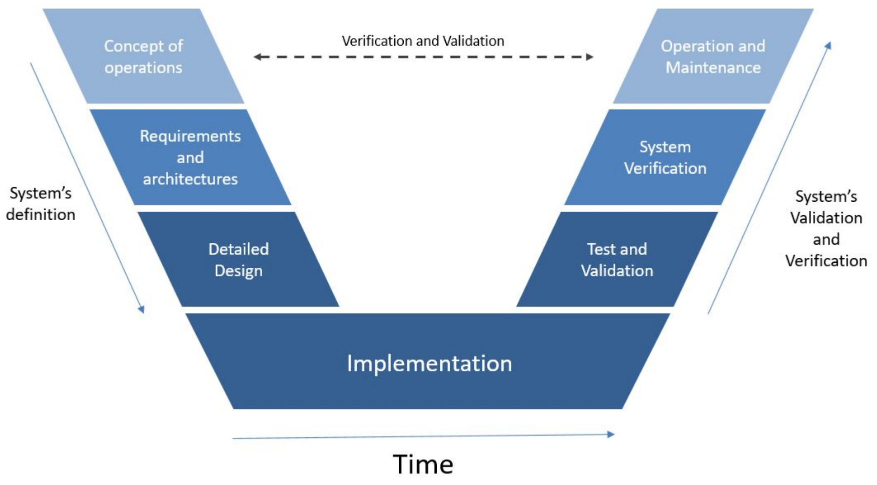

Concept of operations
Verification and Validation
Operation and Maintenance
Requirements and architectures
System Verification
System's definition
System's Validation and Verification
Detailed Design
Test and Validation
Implementation
Time

Figure 1.2 V-model of MBD

1.3 System Verification

The principal validation methods for cyber-physical systems are simulation, testing, deductive verification, and model checking. Simulation and testing both involve making experiments before deploying the system in the field. While simulation is performed on an abstraction or model of the system, testing is performed on the actual product. In both cases, these methods typically inject signals at certain points in the system and observe the resulting signals at other points. For software, simulation and testing usually involve providing certain inputs and observing the cor-

responding outputs. These methods can be a cost-efficient way to find many errors. However, checking all of the possible interactions and potential pitfalls using simulation and testing techniques is rarely possibles.
Model checking is a technique generally used for verifying finite state concurrent systems. One benefit of this restriction is that verification can be performed automatically. It procedure normally uses an exhaustive search of the state space of the system to determine if some specification is true or not. Given sufficient resources, the procedure will always terminate with a yes/no answer. Moreover, it can be implemented by algorithms with reasonable efficiency, which can be run on moderate-sized machines. Applying model checking to a design consists of several tasks, each of which will be discussed in detail later in this thesis.
To study verification methodologies further, let's group verification methodologies into two categories: simulation-based and formal method-based verification. The distinguishing factor between the two categories is the existence or absence of vectors. Simulation-based verification methodologies rely on vectors, whereas formal method-based verification methodologies do not. Another way to distinguish simulation-based and formal method based verification is that simulation is input oriented (for example, the designer supplies input tests) and formal method verification is output oriented (for example, the designer supplies the output properties to be verified). There is a hybrid category called Semi-Formal Verification that takes in input vectors and verifies formally around the neighborhood of the vectors. Because the semi-formal methodology is a combination of simulation-based and formal technology, it is not described separately here.

1.3.1 Simulation-Based Verification

The most commonly used verification approach is simulation-based verification[5]. As mentioned earlier, simulation-based verification is a form of verification by redundancy. The variant or the alternative design manifests in the reference output. During simulation-based verification, the design is placed under a test bench, input stimuli are applied to the test bench, and output from the design is compared with reference output. A test bench consists of code that supports operations of the design, and sometimes generates input stimuli and compares the output with the reference output as well. The input stimuli can be generated prior to simulation and can be read into the design from a database during simulation, or it can be generated during a simulation run.

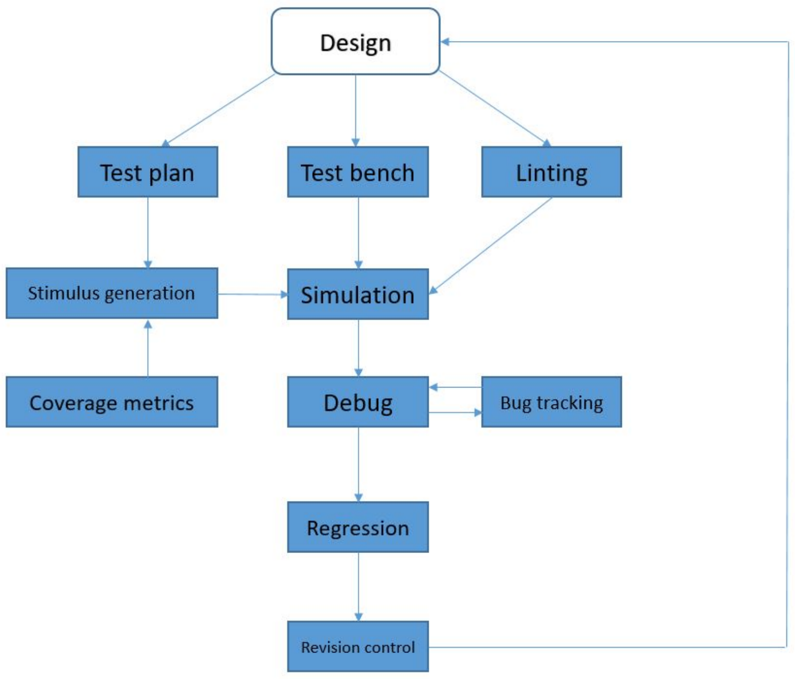


Figure 1.3 Simulation-Based Verification Diagram

1.3.2 Formal Method-Based Verification

The formal method-based verification methodology differs from the simulation-based methodology in that it does not require the generation of test vectors; otherwise, it is similar. Formal verification can be classified further into two categories: equivalence checking and property verification. In this thesis, the property verification is highlighted, especially the model checking techniques. Property checking takes in a design and a property which is a partial specification of the design, and proves or disproves that the design has the property. A property is essentially a duplicate description of the design, and it acts to confirm the design through redundancy. A program that checks a property is also called a model checker, referring the design as a computational model of a real circuit.

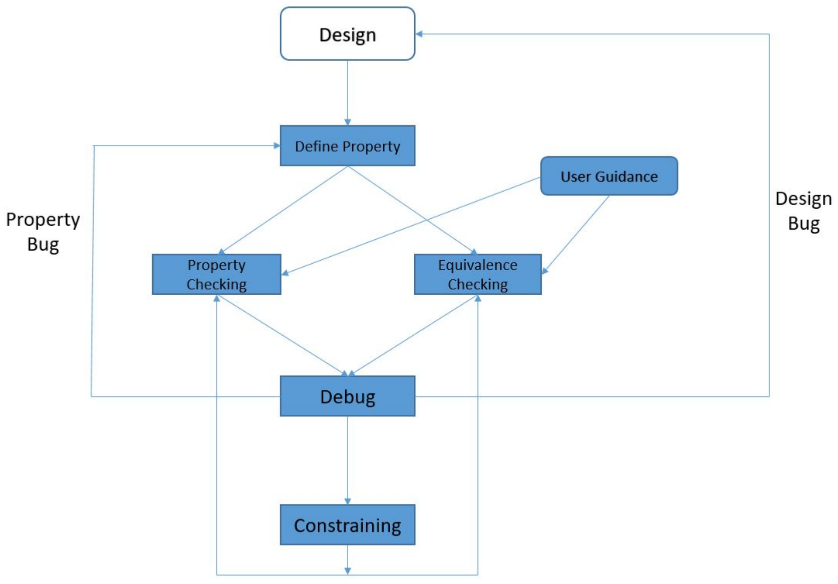


Figure 1.4 Formal Method-Based Verification Diagram

1.4 State of the Art

1.4.1 Formal Methods

Static analysis is used to efficiently compute approximate but sound guarantees about the behavior of a program without executing it [6]. Abstract interpretation relates abstract analysis to program execution [7] and can be used to compute invariants [8], [9], [10]; these approaches have been applied in CPS, including in flight control software [11] and outer space rovers [12]. In general, the efficiency and quality of static analysis tools have reached a level where they can be practically useful in locating bugs that are otherwise hard to detect via testing. However, for mission-critical CPS applications (which may contain millions of lines of code that interact in complex ways with a physical world), existing industry static analysis tools either do not scale well (e.g., [13]) or tend to introduce many false positives (e.g., [14], [15], [16]).

Theorem proving has been applied in deductive verification [17], [18], where validity of the verification conditions are determined. Theorem provers have also been used for verifying hybrid systems [19], [20]. Isabelle/HOL [21] has been used to formally verify the kernel piece of seL4 [22], which is the foundation OS for a highly secure military CPS application. This work shows that, with careful design, a (critical component of a) complex CPS can be formally verified by the state of art theorem prover. However, the requirement for human intervention and high costs (the total effort for proof was about 20 person-years, and kernel changes require 1.5–6 person-years to reverify [23]) makes applying theorem proving impossible for general-purpose CPS applications, which may contain millions of lines of code [24] and require much quicker (and less expensive) changes.

The verification world is overloaded with highly capable model checkers [25], [26], including those that handle real-time constraints [27], parametric constraints [28], stochastic effects [29], and asynchronous concurrency with complex and/or dynamic data structures (though not sharable between concurrent processes) [30], all of which are common in CPS. Model abstraction and reduction can make analysis more tractable (and thus more applicable to CPS) [31], [32], [33], however, error bounds are usually unquantified, which makes the verification unsafe. While model checking allows verification to be fully automated, in addition to issues such as state-explosion, complexity in property specification, and inevitable loss of representativeness [34], CPS exhibits bugs that crop up only at run-time based on the physical state of the deployment world; such bugs cannot be captured by model checking

alone.
It is exceedingly difficult to prove properties of CPS automatically because of the disconnect between formal techniques for the cyber and well-established engineering techniques for the physical [35], [36]. In recent works [37], [38], physical and software processes are modeled and composed together either as timed automata or hybrid automata, and the compositional models are verified by model checkers against correctness properties.

1.4.2 Run-Time Verification

In run-time verification, correctness properties specify all admissible executions using extended regular expressions, trace matches, and others formalisms [39]. Temporal logics, especially variants of LTL [40] are popular in runtime verification. However, basic temporal logics do not capture nonfunctional specifications that are essential in CPS (e.g., timeouts and latency) and lack capabilities to deal with the stochastic nature of many CPS applications [41], [42], [43].
Probabilistic temporal logics such as probabilistic computation tree logic (PCTL) [44] and continuous stochastic logic (CSL) [45] have been introduced two decades ago to specify probabilistic properties, and a subset of CSL [41] and a CSL compatible language [43] are used to monitor probabilistic properties at runtime. These temporal logics lack the capacity to monitor the continuous nature of the physical part of CPS applications. In [46], a monitor is created for stochastic hybrid systems and a monitorability theorem is provided. However, there is little discussion of whether the monitor will impact the system's functional and nonfunctional behaviors. In [47], an efficient runtime assertion checking monitor is proposed for memory monitoring of C programs. This noninvasive monitoring is well suited to mission-critical and time-critical CPS applications.
In summary, the state of art in run-time verification can potentially provide a great supplement to formal methods and traditional testing in CPS. Many opportunities remain to make run-

time verification more suitable to the idiosyncrasies of CPS and approachable to CPS developers. For instance, aspect-oriented monitoring tools [48] are less intrusive, and their adaptation to CPS run-time verification may prove more approachable for developers.

1.4.3 Model-Based approach

In this category, we talk about model-based verification, which uses formal models to enable (automatic) testing of CPS applications [49]. We also include simulation-based techniques aimed at the model analysis of CPS applications [50], [51], [52]. Modeling real-time components has been decomposed into behaviors, their interactions, and priorities on them; reasoning can then occur layer by layer [53], [54]. In general, such approaches allow the verification of all system layers from the correctness proof of the lower layers (i.e., gate-level) to the verification procedure for distributed applications; such an approach has been used to verify automotive systems, a key exemplar of CPS [55]. The practicality and costs of development associated with these approaches are still unknown. While there are many computational and network simulators that many software engineers may be familiar with, in the CPS domain, one of the most relevant systems is Simulink, which is widely deployed in the automotive industry and other mission critical domains (e.g., avionic applications) [56]. In [57], A MATLAB toolbox called S-Taliro is created to systematically test a given model by searching for a particular system trajectory that falsifies a given property written in a temporal logic. However, S-Taliro suffers from a memory explosion problem when a system contains larger specification formulas. MATLAB also provides Simulink design verifier (SDV) [58] as an extension tool-set to perform exhaustive formal analysis of Simulink models. Modelica_Requirements library [59] is an extension of Modelica language and implemented to check formally the defined requirements by simulation.

1.5 Thesis Focus and Contributions

In this thesis, our main focus is on control systems modelled in Modelica which is a widely used tool in the area of control engineering. In order to overcome the aforementioned limitations in improving the resources to verify the CPS, we devised and implemented the two following approaches: (i) a state space definition based on the steady state analysis and (ii) verifying the resulted finite transition system with a model checking tool.

1.6 Outline

Chapter 2 illustrates the notions and definitions that are used within this thesis. Chapter 3 presents our modeling approach for the verification purpose. Especially, how the investigated techniques of the steady state analysis and the transition system definition allow the construction of the finite transition system for the verification phase. Chapter 4 introduces the adopted verification approach and the model checking tool. Chapter 5 presents the case study of the power management systems to show the feasibility of the proposed approach. Chapter 6 draws conclusions and outlines future work.

Chapter 2

Preliminaries

2.1 Synchronous Models

Synchronous Reactive component:
A functional component produces outputs when supplied with inputs, and it can be mathematically described using a mapping between input and output values. A reactive component, in contrast, maintains an internal state and interacts with other components via inputs and outputs in an ongoing manner. A synchronous reactive component C is described by

- a finite set I of typed input variables defining the set Q_I of inputs,
- a finite set$O$ of typed output variables defining the set Q_O of outputs,
- a finite set S of typed state variables defining the set Q_S of states,
- an initialization *Init* defining the set $[[Init]] \subseteq Q_S$ of initial states, and
- a reaction description React defining the set $[[React]]$ of reactions of the form $s \xrightarrow{i/o} t$, where s, t are states, i is an input, and o is an output.

Executions of a component:
The operational semantics of a component can be captured by defining its executions. To execute a component, we first initialize all the variables to obtain an initial state. The execution then proceeds for a finite number of rounds. In each round, values for input variables are chosen. Then the code in the reaction description of the component is executed to determine its output and updated state.

State machines:
State machines are commonly used for describing the behavior in model-based design. It is a model of a computational system, consisting of a set of states, a set of possible inputs, and a rule to map each state to another state, or to itself, for any of the possible inputs.

2.2 Properties of Components

Finite-state components:
In many CPS applications, it suffices to consider types with only finitely many values. A synchronous reactive component C is said to be finite-state if the type of each of its input, output, and state variables is finite.

Combinational Component:
A synchronous reactive component C is said to be combinational if the set of its state variables is empty.

Event-triggered Component:
For a synchronous reactive component $C = (I, O, S, Init, React)$, a set $J \subseteq I$ of input variables is said to be a trigger if:

- every input variable in J is of type event;
- every output variable either is latched or is of type event; and

- if i is an input with all events in J absent (that is, for all input variables $x \in J, i(x) = \perp$), then for all states s , if $s \xrightarrow{i/o} t$ is a reaction, then s = t and $o(y) = \perp$ for every output variable y of event type.

A component C is said to be event-triggered if there exists a subset $J \subseteq I$ of its input variables such that J is a trigger for$C$.

Deterministic Component:
A synchronous reactive component C is said to be deterministic if:

- C has a single initial state, and
- for every state s and every input i, there is precisely one output o and one state t such that $s \xrightarrow{i/o} t$ is a reaction of C.

Continuous-time Component
A continuous-time component H has a finite set I of real-valued input variables, a finite set O of real-valued output variables, a finite set S of real-valued state variables, an initialization *Init* specifying a set [[*Init*]] of initial states, a real-valued expression h_y over $I \cup S$ for every output variable $y \in O$, and a real-valued expression f_x over $I \cup S$ for every state variable $x \in S$. Given a signal I over the input variables I, a corresponding execution of the component H is a differentiable signal S over the state variables S and a signal O over the output variables O such that

- $S(0) \in [[Init]]$;
- for every output variable y and time *t,y(t)* equals the value of h_y evaluated using the values *I(t)* for the input variables and *S(t)* for the state variables; and
- for every state variable x and time t, the time derivative (d/dt) x(t) equals the value of fx evaluated using the values I(t) for the input variables and S(t) for the state variables.

Linear Component A continuous-time component H with input variables$I$, output variables O, and state variables S is said to be a linear component if:

- for every output variable y, the expression h_y is a linear expression over the variables $I \in S$; and f_x is a linear expression over the variables $I \in S$.

2.3 Safety Requirements

Transition System
A transition system is specified using variables whose valuations describe possible states of the system. The initialization describes the initial values for each of the system variables. Transitions of the system describe how the state evolves and are typically specified using a sequence of assignments and conditional statements that update the state variables, possibly using additional local variables. Following the convention analogous to the definition of synchronous reactive components, we use *Init* to denote the syntactic description of the initialization, with an associated semantics [[*Init*]] denoting the corresponding set of initial states. Similarly, *Trans* denotes the syntactic description of the transitions, and the associated semantics [[*Trans*]] is a set of pairs of states. A transition system T has:

- a finite set S of typed state variables defining the set Q_S of states,
- an initialization *Init* defining the set [[*Init*]] $\subseteq Q$ of initial states, and
- a transition description Trans defining the set $[[Trans]] \subseteq Q_S \times Q_S$ of transitions between states.

Reachable States of Transition System
An execution of a transition system T consists of a finite sequence of the form $s_0, s_1, ...s_k$, such that:

- for $0 \leq j \leq k$, each s_j is a state of T,
- s_0 is an initial state of T, and
- for $1 \leq j \leq k, (s_j - 1, s_j)$ is a transition of T.

For such an execution, the state s_k is said to be a reachable state of T.

Invariant of Transition System
For a transition system T, a property φ of T is an invariant of T if every reachable state of T satisfies φ.

2.4 Simulation

Simulation Model
A simulation model M is a formal description of the Cyber-Physical System (CPS) to verify (e.g., Power Grid). In particular, the model is written in the language of the simulator being used.

Simulator
A simulator is a software tool that is able to simulate the behaviour of the given time-variant CPS model M to verify.

Simulation Scenario
A simulation scenario (or simply scenario) is a sequence of disturbances of the environment model in which the CPS operates in.

Simulation Interval
A simulation interval T indicates a fixed amount of simulation seconds. Note that the amount of simulation seconds is usually different from the elapsed time the simulator takes to actually simulate them. In fact, the elapsed time needed to simulate a given simulation interval T may be higher or lower than T , depending on the complexity of the simulation model.

Simulation Setting
A simulation setting $S = (T, M, \sigma, opts)$ identifies the simulation interval, the model M to simulate, the operating environment scenario of the model σ, and the option *opts* of the simulation process (type of the equation solvers, size of the integrator step size,...).

Simulation outputs
The output of a single simulation is a sequence of states also called state vector. Each state is defined by the values of its state variables. The size of the states sequence is computed by the division of the simulation interval and the integrator step size.

2.5 Verification

Abstract State Machines
Proposed by Gurevich [60], abstract state machines (ASM), also called evolving algebras, form a specification language in which the notions of state and state transformation are central. A system is described in this formalism by the definition of states and by a finite set of state transition rules, which describe the conditions under which a set of transformations (modifications of the machine's internal state) take place. These transitions are not necessarily deterministic: the formalism takes into account configurations in which several transitions are eligible for a certain state of the machine.

Automata-Based Modeling
A different class of transition systems used for specification purposes are automata [61]. In this case it is the concurrent behaviour of the system being specified that stands at the heart of the model. The main idea is to define how the system reacts to a set of stimuli or events. A state of the resulting transition system represents a particular configuration of the modelled system. This formalism is particularly adequate for the specification of reactive, concurrent,

or communicating systems, and also protocols. It is however less appropriate to model systems where the sets of states and transitions are difficult to express.

Property

It is and expression defined in temporal logic or computational tree logic allowing the formal verification techniques to demonstrate that a system design is correct with respect to its specification. We can distinguish two type of properties: Safety and Liveness.
Informally, a safety property expresses that *"something bad will not happen"* during a system execution. In real case scenario, the safety properties could be : "The power plant will never blow up", "The reactor temperature will never exceed 100 degree" or "As long as the key is not in the ignition position, the car won't start." A liveness property expresses that eventually *"something good must happen"* during an execution. The distinction of safety and liveness properties was motivated by the different techniques for proving those properties.

Chapter 3

CPS Modeling for Formal-method Based Verification

Basically, SBV consists of simulating both hardware and software components at the same time, in order to analyze all the relevant scenarios that the CPS being verified must be able to safely cope with. Due to the risks associated with errors in CPSs, it is extremely important that SBV performs an exhaustive analysis of all the relevant event sequences (i.e., scenarios) that might lead to system failure. SBV does not have a formal language to specify both the environment where the system will operate, and the safety properties it is necessary to satisfy. As a result of these specifications, the formal verification is generally done after the simulation process. A successful SBV process can guarantee that the CPS will behave as expected, assuming that all and only the relevant scenarios are specified, but cannot verify all possible scenarios. Thus, the need of using the formal-methods is a necessary task.
Starting from a given input scenarios, we generate the corresponding simulation process. The first main idea is to implement the capability of the simulators to store intermediate states and use them to initialize the next simulations, thus avoiding redundancy

that is the variables values describing the same system's state and removing the need to explore common paths of scenarios multiple times.
The second idea is to discretize the overall simulation outputs in order to get them admissible by the model checking verification. Since many scenarios can lead to the same intermediate state, many simulations which explore the same states multiple times can be considered redundant thus making the size of the state space of the system larger and slowing the verification process.
For example, let's consider two different input scenarios d and d' when simulated, give the same outputs. In this case, we store only once the resulting states, but with two different labels.
Hence, two main research questions are formulated. How can we determine if many simulations outputs could be represented by the same state? How can we be sure we have reached all the possible system's states in order to construct the finite transition system for the verification process?

3.1 Methodology

The adopted methodology is divided in two main steps: First the analysis of the steady-state of the system and second the definition of the state space.

3.1.1 Analysis of the steady-state

Several techniques were developed to analyze the steady state for example on [62] where is consider the method of harmonic balance as a general approach to converting a set of differential equations into a nonlinear algebraic system of equations or [63] which the system is interpreted in terms of augmented linear time-invariant (LTI) networks whose frequency-domain solution directly provides the harmonics of the steady-state responses.
Therefore, the main goal of the steady-state analysis in our approach is to extend the existing methodology by introducing an

improved modeling technique to construct the set of states composing the transition system that is given then in input to the model checker in the verification phase.
Basically, a system is in a steady state if the variables (state variables) which define the behavior of the system or the process are unchanging in time. In this thesis, the steady state refers to an equilibrium condition of a circuit that occurs when the effects of transients are no longer important, in other words when the oscillations of a signal are no longer irregular. As said previously, the outputs of a single simulation is defined by a sequence of states describing the behavior and the actions of the system, thus the aim of the steady-state analysis is to recognize a cycle of states in which the system reaches an equilibrium condition. The problem occurs when the number of states composing a cycle might be high considering all the possible simulation scenarios, thus increasing the number of computations and slowing the verification process. In this case, the values of the state variables composing the cycle are quantized in order to *choose* a single or few representative states for every cycle, depending on the desired precision factor. To this end, two methods are proposed: a time-domain and a frequency-domain analysis.
The main goal of the time-domain steady-state methods are to find a steady-state solution with less computational costs than the another analysis method requires, with respect to time. The analysis is done by observing and analyzing the plots of the state variables in function of time. We define straightforwardly the time interval in which the system have reached the steady-state by settling a starting time point and an ending time point of the steady-state period. This will reduce significantly the complexity and the costs of the steady-state computation. However, depending on the simulation scenarios, the system can reach the steady-state in different time points which could create an ambiguity on the selection of the steady-state period. To this end, a function was implemented offering the possibility to specify the time interval and the number of representative states for each scenario, the output of this function is a single or a set of representative states.

On the other hand, the aim of frequency-domain methods refers to the analysis of states with respect to frequency, rather than time. The computational costs using this methods are higher than the time-domain method but the accuracy is guaranteed. The analysis is done by counting the redundancy of a single state in a sequence of states. For example, given a state s belonging to a sequence of states given in output by a simulation process S, we start by running through S and verify which state s has the highest frequency factor, strictly speaking the state which is mostly redundant. The cycle of steady-state will be defined as the states existing between the last two points in which s has been encountered, choosing this two points we are sure that the system is not on a transient state. Once the cycle is found, we quantize it in order to get the representative set of states.
In case of selecting a single state as representative, the quantization function gives in output the average of the values of the state variables composing the cycle. If it's a set that represents the cycle, the choice could be the maximum, minimum and the average of states variables values, depending on the precision factor.

3.1.2 Finite Transition System Definition

The main goal of this computation is to *bound* the number of states composing the transition system. The idea is to compare the resulted states in order to identify if different simulation scenarios can eventually lead to the same states. This process requires two operations, *storing* a resulted state and *initializing* a simulation using the values of the state variables of a stored state. The output of this process will be a set of states Q and a set of transitions T between the states existing in Q. By way of explanation, T will contain the label of the initialization state, the environment scenario and the label of the resulted state from the simulation. The label of a state is a notation referring to certain values of state variables belonging to some state stored in Q (i.e : p $\xrightarrow{SC}$ q) where p is the label of the initialization state, q is the label of the resulted state and SC is the scenario .

1. FOR ALL environment scenarios, SIMULATE without initializing the state variables (the values of the state variables are null).
2. STORE the resulted representative states in Q.
3. INITIALIZE the next simulations with the representative states resulted from the previous simulations and SIMULATE FOR ALL environment scenarios.
4. CHECK if the resulted representative state belongs to Q.
   - IF YES, ADD the label of the initialization state, the environment scenario and the label of the resulted state in T.
   - IF NO, STORE the resulted state in Q, then ADD the label of the initialization state, the environment scenario and the label of the resulted state to T.
5. ITERATE 3 and 4 until no new state is found.

3.2 Steps of the Algorithm

We suppose that a state is defined by a single state variable. Let's start by defining the time-domain function where the steady-state analysis and the quantization process are implemented (Figure 3.1) . The function takes in input the sequence of states q resulted from the simulation, the value of the starting time point t_s and the value of the ending time point t_f of the steady-state interval. The variable *time* is an array containing the numerical integration steps in the simulation time. The size of this array is the same as the size of the states vector, it is due to the generation of a state during every integration step. We need to extract first the indexes from the time array when time is equal to t_s and t_f (line 2 to line 8). These two indexes are used to select the cycle of states in the states vector(line 10 to line 12). In this function a single representative state was chosen.

Figure 3.1 Time-domain steady state analysis Algorithm

```
**float** $quantize(\,q\,,\ t_s\,t_f)\,\{$
**for** ( $i$ **in** **range**(0, **length**($time$), 1)) **do**
    **if** $time[i] == t_s$ **then**
        $start = i$
    **else if** $time[i] == t_f$ **then**
        $end = i$
    **end if**
**end for**
$W = 0$
**for** ( $k$ **in** **range**($start$, $end$, 1)) **do**
    $W += q[k]$
**end for**
**return** $average(W,\ end - start)$
$\}$
```

The frequency-domain function takes in input the state vector q and the frequency of this state *ssFlag* in the states vector (see Figure 3.2). Running through the state vector, the first operation is to count how much time a state existing in q is encountered (line 3 to line 7), if the counter is greater than the given frequency then we extract the indexes of the last two point *start* and *end* in which the state has been encountered (line 8 to line 13). These two indexes are used to select the cycle of states in the states vector (line 15 to line 19).

Figure 3.2 Frequency-domain steady state analysis Algorithm

```
float $quantize(q, ssFlag)${
$count, start, end, W = 0$
for ( $i$ in range(0, length($q$), 1)) do
    for ( $j$ in range(0, length($q$), 1)) do
        if $q[i] == q[j]$ & $i < j$ then
            $count++$
        end if
        if $count > ssFlag$ then
            $start = i$
            $end = j$
            break
        end if
    end for
end for
for ( $k$ in range($start$, $end$, 1)) do
    $W+ = q[i]$
end for
return $average(W, end - start)$
}
```

In the algorithm on the figure 3.3, the process of the construction and reduction of the transition system is described. The inputs are a set Q that contains the states composing the transitions system and T the transitions between those states. the aim of the function *next()* is to : First, simulate a model M assigning as initial state the states existing in Q (if $Q = \emptyset$ the simulation will start without initialization) for all given environment scenarios *Env_Scenarios* and quantize the results of the state sequences q resulted from the simulation, this quantized state will be stored in *q'*. Second, the set of the transitions will be updated with the initialization state p, the scenario SC and the resulted state *q'*. Finally, a verification is done in order to check whether a resulted state has been already found in previous simulations.

Figure 3.3 Definition of the function next()

```
1: set $next(T, Q)$ {
2: $Z \leftarrow \emptyset$
3: for all ($p \in Q \,\&\, SC \in Env\_Scenarios$) do
4:     $q \leftarrow simulate(M, SC, simInterval, p)$
5:     $q' \leftarrow quantize\,(q, t_s, t_f)$
6:     $T \leftarrow T \cup \{[(p\,,\,q'), SC]\}$
7:     if $q' \notin Z$ then
8:         $Z \leftarrow Z \cup \{q'\}$
9:     end if
10: end for
11: return $Z$
12: }
```

The algorithm (figure 3.4) describes the overall process. First, we start by declaring the set containing the system's states and the transitions. After that, initial simulation for the whole simulation scenarios. The set *Q* and *Q'* are used to save the previous and the actual set of states in each iteration in order to compare them later to check if the resulted states are the same obtained from the previous simulations. The algorithm gives as outputs the set of the finite system's states and the transitions between them, those sets are used to elaborate the transition system.

Figure 3.4 Main algorithm

```
1: $T \leftarrow \emptyset$
2: $Q \leftarrow \emptyset$
3: $Q_0 \leftarrow \{0, \cdots, 0\}$
4: $Q' \leftarrow next(T, Q_0)$
5: while $Q \neq (Q \cup Q')$ do
6:     $Q \leftarrow Q \cup Q'$
7:     $Q' \leftarrow next(T, Q')$
8: end while
9: return $Q\,, T$
```

3.3 Conclusion

In this chapter, the steps of the construction of the transitions system given to the model checking tool were explained. The steady states analysis and the transition system definition technique were implemented in order to discretize the system at hand for the verification purpose. The next chapter describes the proposed verification approach where the model checker tool is defined and the verification methodology is proposed.

Chapter 4

Model Verification

4.1 Model Checking

Model Checking [64] is a technique for verifying finite state systems (FSMs). The advantage of this kind of systems is that verification is completely automatic. The process uses an exhaustive search of a given state space of a system to determine if a specification is true or false. Given enough resources, the process will always finish with a yes/no answer. In addition, it can be implemented by algorithms with modest efficiency, which can be executed on moderate-size machines (not on an average desktop computer).

Even if the restriction to FSMs may be a major disadvantage, model checking is applied to various type of systems. Many controllers are finite state systems, and many communication protocols. Sometimes, systems that are not finite are verified using a combination of model checking and abstraction and induction principles like our approach.

Applying model checking to a design consists of several tasks:

Modeling: Converting the design into a formalism (i.e: transition system) recognized by a model checker. In some case, having limitations of resources (time and memory), the modeling design may require an abstraction in order to remove irrelevant and unimportant details.

Specification: It is mandatory to state the properties that a de-

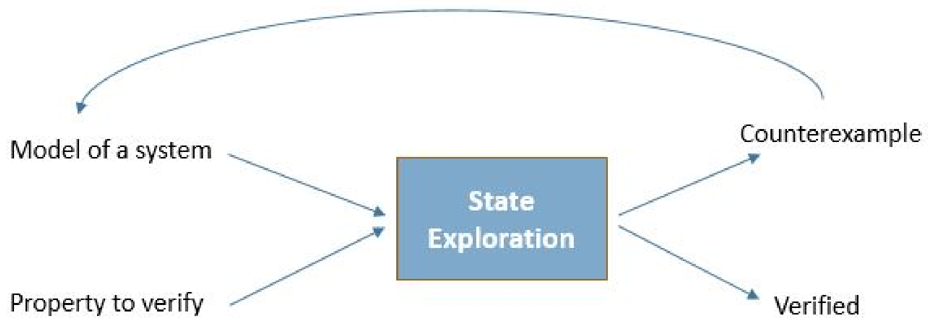


Figure 4.1 Model Checking process.

sign must satisfy before the verification. For CPSs, it is usual to use *temporal logic*, which can define how the behavior of a system evolves over time. The specification is very critical because it is impossible to determine whether a given specification covers all the properties that the CPS should satisfy.
Verification: The verification is not fully an automatic task. In practice, it evolves human assistance. The human expert must analyze the verification results. In case of a *"No answer"*, the expert is provided by an error trace, called also a *counterexample*.

4.2 Model checking tools

Since developing a model checker is a complicated process and since several model checkers already exist, it is easier to choose an existing tool that is created and managed by experts in the model checking field. In this section, we give a brief presentation of 4 model checking software : **Spin**[**65**], **NuSMV**[**66**], **FDR2**[**67**], **Alloy**[**68**].
Spin: It was one the first model checker developed, starting in 1980. It introduced the classical approach for on-the-fly LTL model checking. Specifications are written in Promela [69] and properties in LTL. An LTL property is compiled in a Promela never claim, i.e. a Büchi automaton. Spin generates the C source

code of an on-the-fly verifier. Once compiled, this verifier checks if the property is satisfied by the model. Working on-the-fly means that Spin avoids the construction of a global state transition graph. However, it implies that transitions are (re-)computed for each property to verify. Hence, if there are n properties to verify, a transition is potentially computed n times, depending on optimizations.
NuSMV: It is a model checker based on the SMV (Symbolic Model Verifier) software, which was the first implementation of the methodology called Symbolic Model Checking described in [70]. This class of model checkers verifies temporal logic properties in finite state systems with "implicit" techniques. NuSMV uses a symbolic representation of the specification in order to check a model against a property. NuSMV is described later on details.
FDR2: It is an explicit state model checker for csp, the well known process algebra. fdr2 can check refinement, deadlocks, livelocks and determinacy of process expressions. It gradually builds the state-transition graph, compressing it using state-space reduction techniques, while checking properties, which also makes it an implicit state model checker.
Alloy : Alloy is a symbolic model checker. Its modeling language is first-order logic with relations as the only type of terms. Basic sets and relations are defined using "signatures", a construct similar to classes in object-oriented programming languages, which supports inheritance. Alloy uses SAT-solvers to verify the satisfiability of axioms defined in a model and to find counterexamples for properties (theorems) which should follow from these axioms.

4.3 Properties expression

Temporal logics have been used to specify properties of concurrent systems since they were first proposed for this purpose around 1976, by Amir Pnueli [40]. Temporal logic is a particular variant of modal logic, where new logical operators are introduced to quantify over a family of objects in the system. In the case of temporal

logic in the simplest form, the object in question is a sequence. We interpret this sequence as possible states of the system being specified, as changed over time by actions. This is the only *"temporal"* aspect about the logic. Time does not appear explicitly in it, and in fact the operators could be applied to any sequence.
The temporal logic is divided in two main categories : First, the Linear Temporal Logic [40] (LTL) and Computational Tree Logic[71] (CTL). In LTL, one can encode formulae about the future of paths, (e.g., a condition will eventually be true, a condition will be true until another fact becomes true, etc). The LTL have four temporal operators:

1. **Next :** The next operator is used to specify that a formula holds at next, that is, at the next position in a path. Given proposition a, formula **X** a holds on a path if it holds at the next state, that is, state at position 1 on the path.

2. **Until :** The until operator specifies that a formula is true until another one is true. There are two parts in the definition of φ **U** ψ :

   - formula ψ must hold at some position on the path.
   - at all previous positions, formula φ must hold.

3. **Eventually :** This operator is noted **F** and sometimes $\Diamond$. Intuitively, **F** φ means that φ must hold somewhere in the future. Formally, **F** φ is defined as **True U** φ.

4. **Globally :** This operator is noted **G** or sometimes $\Box$. This is the dual of the eventually operator. Intuitively , **G** φ means that φ always holds.

The second category of temporal logic is the Computational Tree Logic (CTL) which is a branching-time logic, meaning that its model of time is a tree-like structure in which the future is not determined; there are different paths in the future, any one of which might be an actual path that is realized. The temporal operators are the following:

- The path's quantifiers operators :
  1. **A** φ: φ has to hold on all paths starting from the current state.
  2. **E** φ: there exists at least one path starting from the current state where φ holds.
- The Path-specific quantifiers which are the same as the LTL operators.

The figure below shows how CTL properties can be seen.

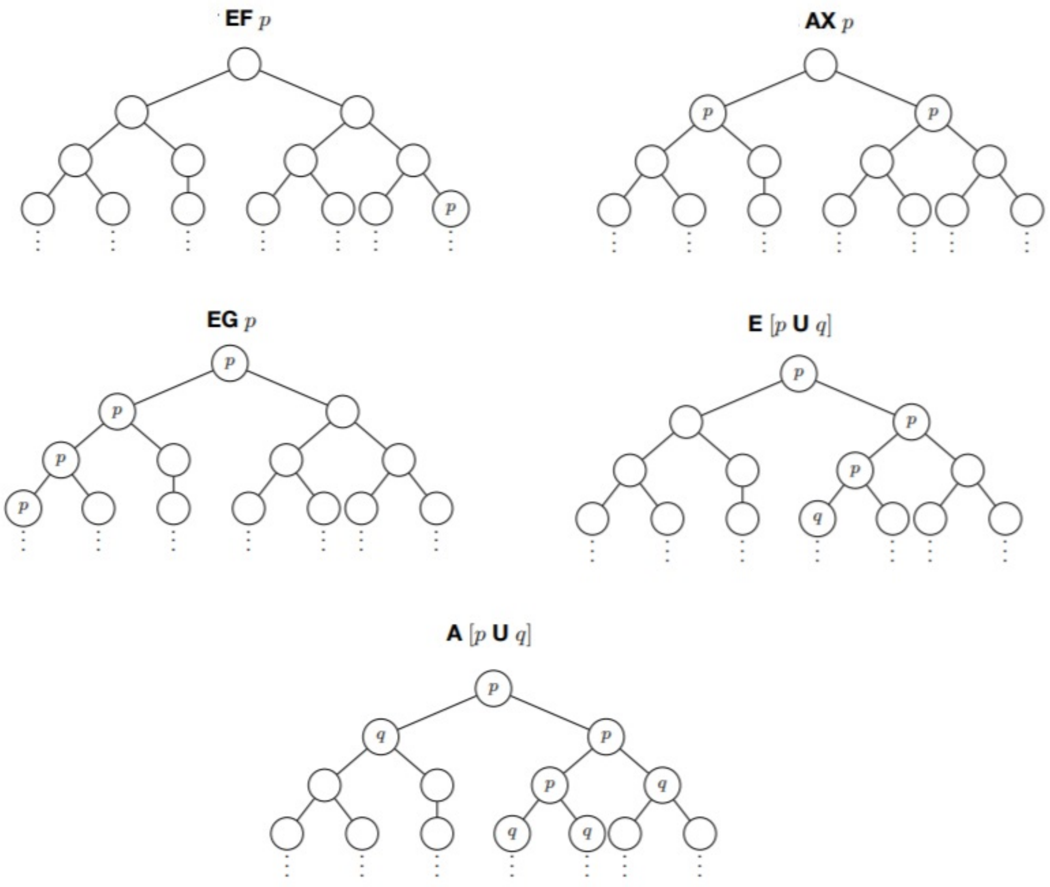


Figure 4.2 CTL Formulas.

4.4 Model Checking Tool: NuSMV

The aim of the previous chapter is to model the system in order to make it acceptable by a model checking tool. To this end, we use NUSMV [66] which is a symbolic model checker tool allowing for the representation of finite state systems, and for the analysis of specifications expressed in CTL and LTL, using BDD-based and SAT-based model checking techniques. The interaction with the user can be done with a textual interface, as well as in batch mode. The tool NuSMV comes with a description language that is used to describe the (finite) transition system modelling the CPS. The description is broken down into modules that can be composed and reused. This modules describe initial values of variables and how they change in each step. NuSMV has also primitives to describe synchronous and asynchronous concurrent computations. The variables can have **boolean** (1 or 0) type, **enumerated** type, finite range of **integers** , or **finite arrays** (NuSMV does not accept the Real type, to this end a conversion into integer is necessary). The keyword **init** is used to describe the initial value if the init value is unspecified the variables can take any value in their type as the initial value. The keyword **next** describes how the value of the variable changes in one step again if the next value is unspecified, then the variable takes any value in its type at the next step. The **case** statement assigns the value associated with the first case condition that is true right now. The example below shows the structure of NuSMV program.

```
MODULE main
  VAR
    request : boolean;
    state   : {ready, busy};
  ASSIGN
    init(state) := ready;
    next(state) := case
                     state = ready & request = 1 : busy;
                     1                           : {ready, busy};
                   esac;
```

Each module can have requirements described in temporal logic. The requirements either described in LTL or in CTL.
CTL specifications are described using the keyword SPEC while LTL specifications are described using the keyword LTLSPEC. In LTL, the logical connectives are as follows: **X** (neXt), **G** (Globally or always), **F** (eventually in the Future), **U** (until), **V** (releases), plus past time operators.

4.5 Verification Methodology

The adopted methodology is divided in two main operations: First, a verification is performed on the finite transition system generated by the CPS modeling phase, this transition system is composed by representative states each state variable assumes a single value, the set of the input symbols are represented by the environment scenario. All the states could be the initial state depending on the sequences of input symbols, thus the execution of this transition system can consider many execution paths. To this end, the properties can be expressed only in CTL using the path-quantifiers and path-specific operators. Let's suppose that S is the transition system and φ is a safety property expressed in CTL, the goal is to verify if $S \models \varphi$. As instance, the property φ could be "As long as the key is not in the ignition position, the car won't start" said in temporal logic **AG** (start $\rightarrow$key). If the property is not violated this means that no further investigation is needed, but if

the property is violated the model checker will produce a counter example, in other words, a finite path that leads to the state in which the property is false or there exists a finite execution that reveals this fact. In this case, we can proceed to the second step of the verification.
Second, in case of erroneous safety property, we need to trace-back the state sequences resulted from the simulation process, that are represented by the *faulty state*. Possibly, this state could not represent only a single state vector generated from a single simulation, but many state vectors generated from multiple simulations. Every sequence of states will be described by a transition system T_j without an input alphabets and with a single execution path. As instance, let's suppose that the property φ is violated within the representative state s and $Q = \{q_0, q_1, ..., q_j, ...q_n\}$ is the set of state vectors q_i resulted from the simulation, where n is the overall number of the executed simulations. If s is the representative state of j vectors, the number of the transitions systems is j as well and φ will be verified among this transition system. We can not represent the obtained state vectors in a single transition system because it is given as a consecutive sequence of states from the simulator and every sequence is proper to the corresponding simulation scenario. Besides of that, there is no explicit transition rules between the states existing in different state vectors.
If $T_k \models \varphi$ where $k \leq j$, the precision chosen for the quantization function could be not adequate to the case at hand. If $T_k \not\models \varphi$, the generated counterexample will give precisely in which state the system became faulty, in this way the designer could retrace exactly the error and modify the design in order to fix the problem.
In the case of $T_k \models \varphi$, a possible verification of liveness properties could be performed since T_k has a unique path of execution, thus the properties expressed in LTL can be verified. The liveness property can be specified in order to investigate further why φ is satisfied within this transition system even if the representative state does not satisfy φ.

4.6 Conclusion

This chapter introduced the two phases of our verification approach and the model checker tool NuSMV. The next chapter shows the feasibility of the modeling and verification approaches with a study case of power management system.

Chapter 5

Case Study

The study of the dynamics of power management plants has been an active research field for more than thirty years, the first pioneering works dating back to the last sixties; a review of basic issues in modelling and simulation of such plants, as well as references to relevant publications in the field can be found, for example in [72].
Nevertheless, the modeling of such systems for automatic verification purposes has not been given such an importance. The adopted approach has been focused primarily on photo-voltaic power plants, driven by many different motivations. In this case, the emphasis is put primarily on safety issues, i.e. on the study of the plant dynamic behaviour in case of failures; this requires the greatest modelling effort to achieve the highest possible accuracy of the results, since the outcome of the simulations has to be used to assess the plant safety in case of accidents, which is obviously a very critical issue. Two main methods can be identified, leading to two main categories of models: simple global plant models, and detailed plant models, often based on modular approaches. In this thesis, the second method has been used by implementing the CPS using the Modelica language described later on. The case study that we are presenting within this work is the Distributed Maximum Power Point Tracking (DMPPT) which is a technique that allows each photo-voltaic (PV) module of an array to operate

in its Maximum Power Point [73]. This is highly desirable, from the efficiency point of view, in presence of mismatching operating conditions. The system is composed of a set of electrical components : Photo-voltaic panels, DC/DC converter, diodes, etc. Our case study validation results are presented in [74].

5.1 Types of DMPPT systems

Functionally, all smart panels can be classified into three categories according to their voltage gain: *buck type, boost type and buck-boost* type. In this section, a summary for the different topologies of the DMPPT converter is elaborated among different DC-DC converter types, highlighting their functional characteristics. Three possible converter types are taken into consideration in turn for their simplicity on the modeling phase and efficiency, each with different strengths and uses.

5.1.1 Boost Converter Smart Panel

The Boost converter smart panel is operating its output voltage above the input MPPT voltage. If the output operating point is at a voltage lower than the maximum power voltage of the PV panel, the converter should be working in go-through mode, which means the output I-V curve is same as the original PV panel I-V curve in such condition. So the Boost converter smart panel's output I-V curve should consist of two parts: one is a constant power curve starting from the voltage of the maximum power operating point M and ending at the crossing point N meeting with the voltage limit; the other part is same as the original PV output I-V curve starting from zero voltage and ending at the voltage of the maximum power operating point. The real strength of the Boost converter smart panel is its simple structure and it can achieve a certain bus voltage with fewer smart panels when they are series connected. Besides, Boost converter smart panels do not even need a current limit design, its maximum output current is equal to I_{sc} of the PV panel. But since part of output I-V curve is composed of the original PV curve, the output curve is very sensitive to irradiance case, so the common current region of maximum power is difficult to ensure when a mismatch case happens in a series connected smart panel based PV string. Thus, the Boost converter smart panel is not suitable for series connection. When the converter's operating point is approaching the region of the voltage limit, the output power of smart panel based PV system may drop dramatically, since the MPPT algorithm can perturb the voltage even beyond the protection limit.

5.1.2 Buck Converter Smart Panel

In a Buck converter smart panel, the output voltage must always be less than the input MPPT voltage, if the output operating point is at a voltage higher than the input maximum power point operating voltage, the converter should be working in go-through mode, which means the output I-V curve is same as original PV panel in this instance. So the output I-V curve of the Buck converter smart panel should consist of three parts: one part is constant power curve starts from point M power operating point, assuming the voltage of the maximum power point of the PV panel labeled as point M, and ending at the crossing point meeting with current limit; another part is horizontal current limit line; the third part is the same as the original PV output I-V curve starting from point M and ending at the open circuit voltage of the original PV panel. In a Buck type smart panel, the simple structure and inherit voltage limit function are major merits. Buck converter smart panels can always guarantee a common maximum power region in series connection even in a mismatch case. What's more, the soft inherit voltage limit curve, is able to prevent a power drop resulting from the MPPT algorithm working within the area around voltage limit. Buck type PV smart panel can increase the output current and its output voltage is limited by the maximum voltage of original PV unit. Moreover, its output current is limited by the rating of the device and the thermal constraints of the PV smart panel.

5.1.3 Buck/Boost Converter Smart Panel

By cascading the boost converter with a buck converter using a common inductor, a four switch Buck/Boost topology can be obtained. The Buck/Boost converter uses Buck and Boost stages to allow the converter to generate an output voltage both above and below the input MPPT voltage, this is very flexible when coping with the variation of sun light irradiation. The input I-V curve (dash line) and output I-V curve (solid line) of the converter: it consists of a constant power curve and current limit and voltage limit. The common MPPT region of the Buck/Boost converter mainly depends on its limit rating, so from this we can conclude that the Buck/Boost converter is suitable for both parallel connections and series connections. The advantage of a Buck/Boost converter is its wider output maximum power range and better ability to handle greater amount of system mismatch cases. A Buck/Boost type smart converter based PV system enjoys most of the benefits of both Buck type and Boost type. A Buck/Boost converter includes two conversion stages, so the increasing quantity of switches and flexibility may come at the cost of efficiency reduction, control complexity and possible size and cost increases relative to a single-stage converter such as the Buck and Boost converter. Also the power drop issue caused by a sharp voltage edge still exists because the voltage limit design of the Buck/Boost converter smart panel is as same as the Boost converter smart panel. Boost type PV smart panel can boost the output voltage higher comparing with input MPP voltage and its output current is inherently limited by the characteristic of PV panel. Boost type PV smart panel can be divided into two groups based on different voltage conversion ratio. In this thesis, Buck/Boost type PV smart panel is preferred which normally operates at nearly unit conversion ratio with high efficiency and low cost. However, such boost PV smart panel needs to be connected in series to achieve high DC bus voltage. The output voltage of Boost type PV smart panel is limited by the rating of the device and capacitor used in the PV smart panel.

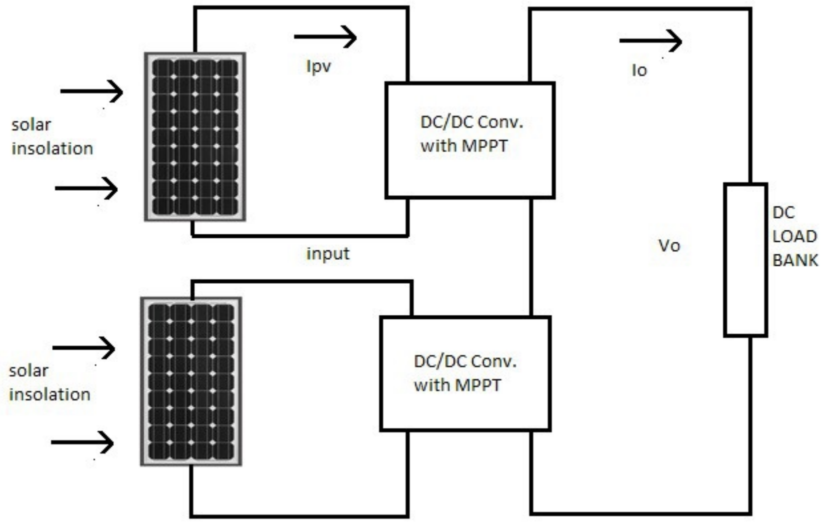


Figure 5.1 Distributed Maximum Power Point Tracking circuit

5.2 Cyber-Physical Energy System

Our flow consists on parameterizable models for the Physical System (PS), the Control Algorithm (CA), and the Computational System (CS), they will be instantiated during the simulation process, which is described later on.

5.2.1 Physical Modeling

Physical modeling deals with the expression of the physical system consisting of plant and actuators in mathematical or logical terms. Continuous dynamics modeling is required since we want to know how the PS regulates the production of energy in function of the irradiation variability. In the modeling of the DMPPT, the model of the overall circuit is composed by a set of physical components: the solar cells, the photo-voltaic panel, a Buck/Boost converter and a dc/ac inverter. We use an equation-based modeling approach to express the physical dynamics of the circuit. As usual, the output current I of each solar cell is computed by solving the Kirchhoff's law:

$$I = I_{ph} – I_r \times q \times e^{(V+I\times R_s)/(\eta\times k\times T)} – 1 – V + I \times R_s \times R_p \quad (5.1)$$

where V is the output voltage of the cell, I_{ph} is the photo-current, I_r is the saturation current, R_s is the resistance in series, R_p is the resistance in parallel, q is the electrical charge (1.6 $\times$ 1019 C), η is the p-n junction quality factor, k is the Boltzmann constant (1.38 $\times$ 1023 J/K), and T is the temperature (in Kelvin degrees).

5.2.2 Control Algorithm

In our approach to model the DMPPT, two controls are adopted: the first control is the Maximum Power Point Tracking (MPPT) based on the Perturb and Observe method[75]. This operates by periodically incrementing or decrementing the output terminal voltage of the photo-voltaic cell and comparing the power obtained in the current cycle with the power in the previous cycle. However, the main control of our design consists in adapting the switching mode (duty cycle) of the dc/dc converter according to the current of the connected panel (I_{pan}) current of the converter (I_{out}). We have modelled each plant in such a way that the software modules of the converter communicates the I_{out} value to the other modules. In this way, each software module of the converter can detect the state in which the others are into and adapt its state accordingly. This is crucial for our verification purposes. More precisely, this control depends on the computation of I_{out} and of I_{pan} and depending on the entered values, the switching mode is determined. The output voltage of the converter (V_{out}) is computed which modifies the switching frequency of the converter by calculating a new value of the MPPT duty cycle.

5.2.3 Computational System

As said in the previous section, the objective of the controller is to provide two main parameters to harvest the maximum energy under different environment conditions through two parameters of the system, the switching mode and the output voltage. According to [76] there are four possible states of the DMPPT system. First the MPPT state, which is the desired working way of the DC/DC converter. In this case, the output voltage is in the range between (V_{max}) and (V_{min}). This protection measure is realized by modeling a blocking diode. The converter in this mode switches according to the computed value of the duty cycle. The second state is the CUT-OFF: this typically occurs when the converters are connected to un-shaded panels and generates much more power than shaded ones; the converter still switches and the output voltage is computed as follows: $V_{out} = V_{max}$. The third state is the PASS-THROUGH: to boost converters that are connected to shaded panels that generates much less power than un-shaded ones; the converter in this mode stops switching and the panel is directly connected in series to the output. The output voltage is computed as follows: $V_{out} = V_{pan}$ – R_s * I_{out}. The fourth state is the BY-PASS: when the panel is heavily shaded and it cannot be connected to the string because it would sink rather than source power. In this mode, the converter bypasses the panel and here too stops switching. The output voltage equation describing this mode is V_{out} = - R_s* I_{out}. In particular, once the control algorithm detects the mode, according to the values of the currents, the formula for computing V_{out} is communicated to the plant, which modifies the value of the duty cycle, this way determining the switching mode. If the control does not detect a change in the values of the currents, then the output voltage remains unmodified.

5.3 Modeling the CPS : Modelica

Modelica [77] is a non-proprietary, object-oriented, equation based language to conveniently model complex physical systems containing, e.g., mechanical, electrical, electronic, hydraulic, thermal, control, electric power or process-oriented sub-components. Its definition is property of a no-profit institution (the Modelica Association), composed by tool vendors and users, contributing to the development of the language and of a suite of standard model libraries.

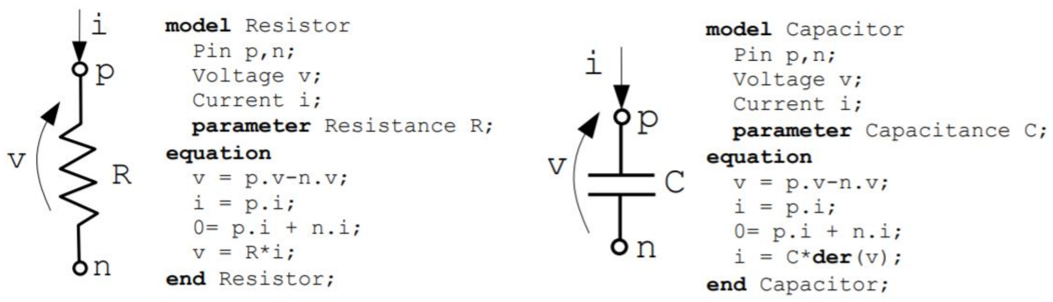
5.3.1 Acausal Modeling

Acausal modeling is a declarative modeling style, meaning modeling based on equations instead of assignment statements. Equations do not specify which variables are inputs and which are outputs, whereas in assignment statements variables on the left-hand side are always outputs (results) and variables on the right-hand side are always inputs. Thus, the causality of equation-based models is unspecified and becomes fixed only when the corresponding equation systems are solved. This is called acausal modeling. The term physical modeling reflects the fact that acausal modeling is very well suited for representing the physical structure of modeled systems. The main advantage with acausal modeling is that the solution direction of equations will adapt to the data flow context in which the solution is computed. The data flow context is defined by stating which variables are needed as outputs and which are external inputs to the simulated system. The acausality of Modelica library classes makes these more reusable than traditional classes containing assignment statements where the input-output causality is fixed. In summary of acausual modeling:

- The model of each system is given by the constituent equations
- The model formulation is independent of the actual boundary conditions, therefore it is re-usable in different contexts $\rightarrow$ modular approach
- The physical connection between components is represented by connection equations
- The simulation of an aggregate system is a difficult task, because in general it could require some form of symbolic

manipulation of the system of equations, prior to the use of some numerical integration algorithm.

A model can contain variables, parameters, constants, other models, and equations relating them. For instance, the example below shows the resistor and capacitor models.

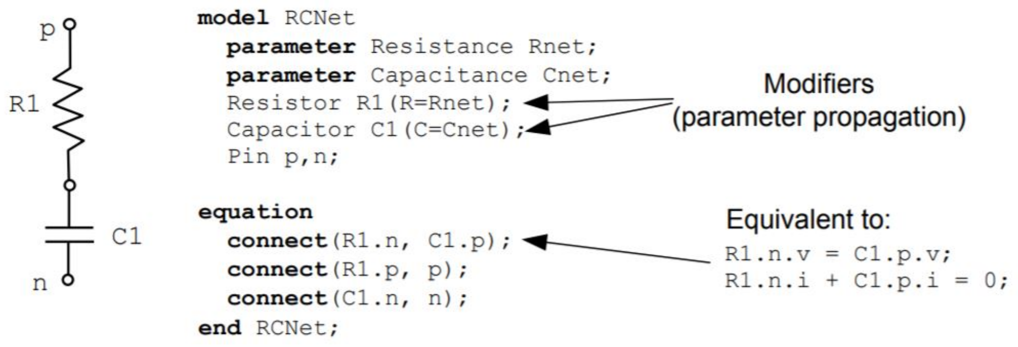


```
model Resistor
  Pin p,n;
  Voltage v;
  Current i;
  parameter Resistance R;
equation
  v = p.v-n.v;
  i = p.i;
  0= p.i + n.i;
  v = R*i;
end Resistor;
```

```
model Capacitor
  Pin p,n;
  Voltage v;
  Current i;
  parameter Capacitance C;
equation
  v = p.v-n.v;
  i = p.i;
  0= p.i + n.i;
  i = C*der(v);
end Capacitor;
```

Figure 5.2 Resistor and Capacitor models in Modelica

It is possible to define models containing other models for example the use of the models defined above in order to build a simple circuit :



```
model RCNet
  parameter Resistance Rnet;
  parameter Capacitance Cnet;
  Resistor R1(R=Rnet);
  Capacitor C1(C=Cnet);
  Pin p,n;

equation
  connect(R1.n, C1.p);
  connect(R1.p, p);
  connect(C1.n, n);
end RCNet;
```

```
R1.n.v = C1.p.v;
R1.n.i + C1.p.i = 0;
```

Figure 5.3 Simple circuit using the Resistor and Capacitor models

5.3.2 DMPPT Models:

Main Model :
The main model is Dmppt is described below. The configModel

is a static model in which the configuration parameters are defined(cells number per panel, number of panels,etc). The Closed-Loop model contains the PV array, the DC/DC converter and the MPPT controller, we set the desired number of modules and customize the architecture by changing the distribution of the components within the block **equation** by changing the computation of the circuit output voltages and currents V_{out} and I_{out}. The mode-control is done on this model where the current and voltage of each module is reachable. The overall circuit is closed by a simplified model of DC/AC inverter. This approach allows the subdivision of the circuit easily, with less customization, guaranteeing flexibility and re-usability of his modules. The code is available in (https://github.com/driysf/DMPPT-CPS)

```
model Dmppt

  import SI = Modelica.SIunits;


  constant configModel conf;
  constant SI.Resistance R_DCAC = 1.5;
  constant SI.Voltage V_DCAC = 0.6;
  constant SI.Time T_mode = 0.001 ;

  ClosedLoop closed[conf.PANELS_NUMBER];



  SI.Current I_out;
  SI.Voltage V_out;
  SI.Power   P_out;


equation

      for i in 1:conf.PANELS_NUMBER loop

        \\ Control the mode of each converter
         when sample(0, T_mode) then

            closed[i].plant.mode =StateController(closed[i].plant.panel.I,
                                  closed[i].plant.panel.V);

         end when;

         \\the modules are in series
          V_out = sum(closed[i].plant.V_out);
          I_out = closed[i].plant.I_out;

      end for;

               /* Output of the Inverter. BEGIN */
               V_out = I_out * R_DCAC + V_DCAC;
               P_out = V_out * I_out;
                /*Output of the Inverter. END */


      end Dmppt;
```

Physical Models :
The physical part of the system is defined by a set of continuous-time components. The main physical components on this system is the PV array composed of a set of PV cells. The dynamic of the solar cell is determined by the Kirchhoff's law that mathematically describes the I-V characteristic of the PV cell. The calculations of the photo-current and the diode saturation current are necessary. The model hereafter shows how the PV cell has been modelled.
The model PVCell is used to define the photo-voltaic panel model, an array of PVCell object is defined, in this step we configure the panel with the position of the cells (in series or in parallel), the irradiation's value in which the panel operates in taken from a configuration model also called a record and the index of the panel used to identifies the panel. The model above shows how the PV panel has been modelled.

```
model PVCell

  import SI = Modelica.SIunits;
  import k = Modelica.Constants.k "Bolzmann constant";


  constant SI.Charge q = 1.6e-19 "Elementary charge";
  constant SI.Energy E_gap = 1.124 * q "Band gap Energy";
  constant SI.Current I_ph_stc = 7.7 "STC photocurrent";

  constant SI.Temperature T_stc = SI.Conversions.from_degC(25) "STC cell
      temperature";
  constant SI.Irradiance G_stc = 1000 "STC sun irradiance";

  parameter SI.Resistance R_s = 0.11 "Cell serial resistor";
  parameter SI.Resistance R_p = 148  "Cell parallel resistor";
  parameter SI.Temperature NOCT = SI.Conversions.from_degC(46) "Nominal
      operating cell temperature";

  parameter SI.Temperature T_amb = SI.Conversions.from_degC(20) "Ambient
      temperature";
  parameter Real nu (unit = "1") = 1.0255 "Diode ideality factor";

  parameter Real C_0 (unit = "A/K^3") = 373 "Diode temperature
      coefficient";

  parameter Real alpha (unit = "1/K") = 0.07 "Current thermal
      coefficient";

  SI.Irradiance G "Sun irradiance";

  SI.Temperature T "Cell temperature";
  SI.Current I_0 "Diode saturation current";
  SI.Current I_ph "Photocurrent";
  SI.Current I;
  SI.Voltage V;
  SI.Power P;

equation
  I = I_ph - I_0 * (exp(q * (V + R_s * I) / (nu * k * T)) - 1) - ((V +
      R_s * I) / R_p) "Kirchhof's law";

  I_ph = I_ph_stc * G / G_stc * (1 + alpha * (T - T_stc));

  T = T_amb + (NOCT - SI.Conversions.from_degC(20)) * G / 800;

  I_0 = C_0 * T^3 * exp(-E_gap / (k * T));
  P = I * V;

end PVCell;
```

```
model PVPanel

  import SI = Modelica.SIunits;

  Integer NumPanel;   // Index of the panels


  irradiationConf conf_Irradiance;

  constant configModel conf;

  PVCell cells[conf.CELLS_NUMBER];

  SI.Current I;
  SI.Voltage V;
  SI.Power P;

equation


        // Set the Irradiance to the panels
        for j in 1:conf.CELLS_NUMBER loop
            cells[j].G = conf_Irradiance.panelIrradiance[NumPanel,j];
        end for;


        // Cell in series (voltage summed up and current the same)
          V = sum({cells[i].V for i in 1:conf.CELLS_NUMBER});
        for i in 1:conf.CELLS_NUMBER loop
          cells[i].I = I;
        end for;


        // Power of the panel
        P = I * V;

end PVPanel;
```

According to the DMPPT architecture, the PV panel is connected to a DC/DC Converter also called a power optimizer, the aim of this component is to maximize the energy harvested from solar photo-voltaic systems. This is done by individually tuning the performance of the panel through maximum power point tracking (MPPT) which is implemented in the control part. This Power optimizers can correct for module "mismatch" by allowing each module to function at its maximum power point (MPP) and then converting the energy to the optimal voltage and current for the panel. This enables the entire array to harvest more energy. The MPPT controller adjust the value of the duty cycle based on the current and voltage produced by the PV array, this adjustment change the value of the switching frequency of the converter allowing the array to *track* always the optimal power. In the model above, the implementation of the converter is presented.First, The parameters (sampling time and switching frequency), physical components, physical units and the PV panel model are declared. Second, on the **equation** block, the physical components are assembled together (condenser, inductor, switching mosfet,etc) by defining the mode's equation system. The *Squarewave()* is a function that alternates at a steady switching frequency of the converter between fixed minimum and maximum values, with the same duration at minimum and maximum, it is used to control the pulses of current that can likely causes noise or errors. The variable **dt** is the duty cycle used on the MPPT control. The discrete variable **mode** is responsible of the mode-control described in the sub-section 5.1.3 (Computational System) , the value of this variable is monitored by the mode control discussed later on.

```
model Plant

  import SI = Modelica.SIunits;

   //Overload Voltage Protection
  constant SI.Voltage V_max = 1.5 ;
  constant SI.Voltage V_min = 0.01;

   // Maximum Power Point Power Value
  constant SI.Power Pmpp = 8.5;


  // Parameters of sampling and switching
  parameter SI.Frequency f_s = 5e4;
  parameter SI.Time T_s = 1 / f_s;
  parameter SI.Time T_mode = 0.001;


  PVPanel panel;
  configModel conf;

  parameter SI.Capacitance C1 = 1e-6;
  parameter SI.Capacitance C2 = 1e-6;
  parameter SI.Inductance L = 1e-3;
  parameter SI.Resistance R_D1_on = 0;
  parameter SI.Resistance R_D1_off = 1e6;
  parameter SI.Resistance R_D2_on = 0;
  parameter SI.Resistance R_D2_off = 1e6;
  parameter SI.Resistance R_D3_on = 0;
  parameter SI.Resistance R_D3_off = 1e6;


  SI.Power P_out;

  SI.Voltage V_out;
  SI.Voltage V_D1;
  SI.Voltage V_D2;
  SI.Voltage V_L;

  SI.Current I_out;
  SI.Current I_C1;
  SI.Current I_C2;
  SI.Current I_D1;
  SI.Current I_D2;
  SI.Current I_L;
  SI.Current I_D3;

  SI.Resistance R_D1;
  SI.Resistance R_D2;
  SI.Resistance R_D3;

  Real dt;
  Boolean on;
  discrete Integer mode;


equation
```

```
                      panel.I=  I_C1 + I_L;
                      I_C1 = C1 * der(panel.V); // condensatore
                      panel.V = V_L + V_D1;
                      V_L = L * der(I_L);      // induttanza
                      V_out = V_D1 - V_D2;
                      V_D1 = R_D1 * I_D1;
                      V_D2 = R_D2 * I_D2;
                      I_D2 = I_out + I_C2;
                      I_C2 = C2 * der(V_out);
                      I_L = I_D1 + I_D2;
                      on =  if SquareWave(T_s, dt, time) > 0 then true else
                            false;
                      R_D1 = if on then R_D1_on  else R_D1_off;
                      R_D2 = if on then R_D2_off else R_D2_on;


                      // Overload protection
                      R_D3 = if (V_out >= V_max) then R_D3_on else R_D3_off;


                      V_out = R_D3*I_D3 + ((panel.V * panel.I) / I_out);
                      P_out = V_out * I_out;


                    when sample(0,T_mode) then

                            if mode == 1 then //MPPT

                                  P_out = I_out * (Pmpp / panel.V);


                             elseif mode == 2 then //CUT-OFF

                                  P_out = I_out * (V_max);


                             elseif mode == 3 then  //PASS-THROUGH

                                  P_out = I_out *(panel.V - (R_D2 *I_out));

                            else //BY-PASS

                                  P_out = I_out * (- (R_D2 * I_out));

                            end if;
                    end when;

end Plant;
```

Controllers:
As said before, the model disposes of two main time-triggered (discrete) controllers the : MPPT controller and the mode controller. The problem considered by MPPT technique is to automatically find the voltage or current at which a PV array should operate to obtain the maximum power output under a given solar irradiance. This method involves a perturbation in the duty cycle of the power converter, and P & O a perturbation in the operating voltage of the PV array. In the case of a PV array connected to a power converter, perturbing the duty ratio of power converter perturbs the PV array current and consequently perturbs the PV array voltage. The process is repeated periodically until the MPP is reached. The system then oscillates about the MPP. The oscillation can be minimized by reducing the perturbation step size. However, a smaller perturbation size slows down the MPPT. The model above *Closedloop* in which the sampling process and the function *Controller()* is invoked. This function takes in input the previous and the actual values of the current and voltage in order to produce the value of the new duty cycle. In each step, the new value of the duty cycle is given to the converter to adjust the switching frequency therefore getting the power value closer to the MPP.
The second controller is the mode controller in which the four possible states of the DMPPT are decided. The control is done in function of the panel's current and the MPP current. The function *ModeController()* takes in input the value of the current and voltage (the voltage is used to verify the BY-PASS when the panel is completely shaded in consequence the current and the voltage take negatives or null values)

```
model ClosedLoop


  constant SI.Frequency f_a = 100;
  constant SI.Time T_a = 1 / f_a;

  Plant plant;


  discrete Real x[3]; // the array x is used to save the previous values
                       // of the current and voltage and the delta duty

 discrete Real u[2];    // the array x is used to save the actual values
                        // of the current and voltage

  discrete Real y;

initial equation


  // State variables of the Duty Controller
  x[1] = plant.panel.I;
  x[2] = plant.panel.V;
  x[3] = 0.01;         // initial value of duty cycle


equation


  when sample(0, T_a) then // We sample to implement the "Perturbe &
                          //  Observe" control
    u[1] = plant.panel.I;
    u[2] = plant.panel.V;
    (x, y) = Controller(pre(x), u);

    plant.dt = y;          // new value of the duty cycle to the converter

  end when;
end ClosedLoop;
```

```
// Perturbe & Observe
function Controller

      parameter Real dutymin = 0.05;
      parameter Real dutymax = 0.95;


      input Real x[3];
      input Real u[2];

      output Real xnext[3];
      output Real y;

      constant Real deltaDuty = 0.05;

      Real Pold;
      Real Pnew;
      Real Vpan;
      Real Ipan;
      Real dtc;

      algorithm

              // Values coming from the input
              Ipan := u[1];
              Vpan := u[2];
              Pold := x[1] * x[2]; // Initialization of the old value
                                   //  of Power
              dtc := x[3];


             if (dtc <= dutymin) then
                  dtc := dutymin;

              elseif (dtc >= dutymax) then
                  dtc := dutymax;

             end if;

             Pnew := Vpan*Ipan;

             if (Pnew <= Pold) then

                    deltaDuty := -deltaDuty;

             end if;

            // update the duty cycle value
            dtc := dtc + deltaDuty;


            //set the new values of the voltage, current and duty cycle
            xnext[1] := Ipan;
            xnext[2] := Vpan;
            xnext[3] := dtc;
            y := dtc;

end Controller;
```

```
function ModeController

      import SI = Modelica.SIunits;

      constant SI.Current deltaCurrent = 0.03;
      constant SI.Current Impp = 2.5;
      constant SI.Current Pmpp = 8.5;


      input SI.Current Ipan;
      input SI.Voltage Vpan;


      output Integer mode;



algorithm

       if Vpan > 0.05 then

              //MPPT
              if (Ipan >= Impp - deltaCurrent and Ipan <= Impp +
                 deltaCurrent) then

                             mode := 1;

              //PASS-THROUGH
              elseif Ipan < Impp - deltaCurrent then

                             mode := 3;

              // CUT-OFF
              elseif Ipan > Impp + deltaCurrent then

                             mode := 2;

              end if;

        // BY-PASS
        else
                       mode := 4;

        end if;

end ModeController;
```

5.4 Benchmark:

Computational Infrastructure
The simulation had been executed on a 516-node Linux cluster with 300 GB of distributed storage capacity. This node consists of 2 Intel Haswell 2.40GHz CPUs with 8 cores and 42 GB of RAM, our hardware configuration allows to run 15 simulations simultaneously. The execution time of the overall simulation process is 63 hours.The number of reached states is 1253.

Parameters of the simulation
In our experiment, we modelled two photovoltaic panels, twelve cells per panel, with different levels of irradiation, with 0.01 seconds as sampling time of the MPPT control and 0.05 seconds as sampling time for the converter state control. The table below shows the parameters used to execute the experimental simulations.
The irradiation of a single panel can take one of the following percentage of the nominal irradiation: {0%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 90%, 100%}.

Parameters	**Values**
Nominal Irradiation of the panels	1000 W/m^2
R_s (series resistance)	0.11 Ω
R_p (parallel resistance)	148 Ω
T_{amb}	20 °C
I_{ph_STC}	7.7 A
T_{STC}	25 °C
T_{NOCT}	46 °C
F_s (Switching frequency of the converter)	5^4
T_a(Sampling time of the MPPT control)	$1/F_s$ seconds
T_mode(Sampling time of the mode control)	0.001 seconds
V_{max}	1.5 V
V_{min}	0.01 V
Panel's Number	2
Cells number per panel	12

5.5 Simulating the CPS : JModelica.org

Simulator Description
JModelica.org [78, 79] is an extensible Modelica-based open source platform for optimization, simulation and analysis of complex dynamic systems. The main objective of the project is to create an industrially viable open source platform for simulation and optimization of Modelica models, while offering a flexible platform serving as a virtual lab for algorithm development and research. As such, JModelica.org provides a platform for technology transfer where industrially relevant problems can inspire new research and where state of the art algorithms can be propagated from academia into industrial use. JModelica.org is a result of research at the Department of Automatic Control, Lund University, and is now maintained and developed by Modelon AB in collaboration with academia.
JModelica.org at a glance:

- Model your systems using the object-oriented and equation-based language Modelica
- Solve your complex simulation and optimization problems using state of the art numerical algorithms
- Automate your work in the Python scripting environment Visualize your results

JModelica.org is compliant with the Modelica Standard Library.

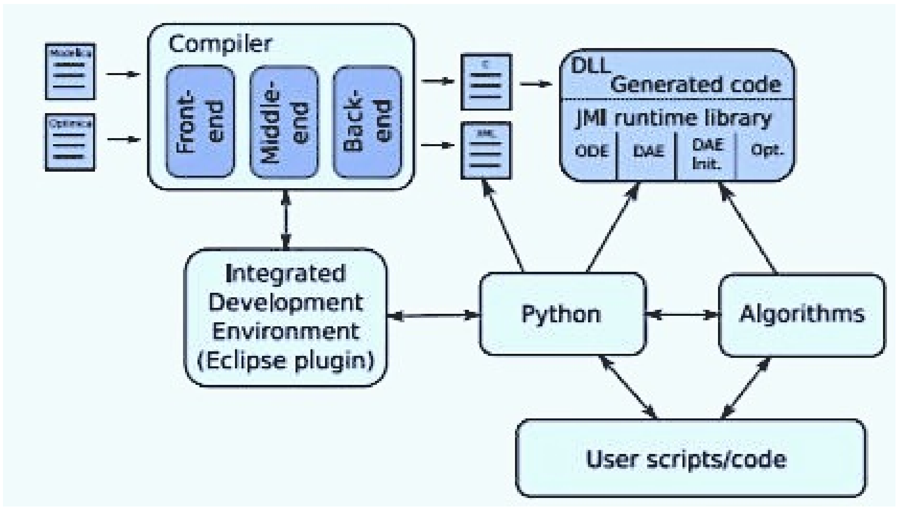


Figure 5.4 JModelica.org architecture

Model Simulation

We simulate our CPS model using the JModelica.org platform, the configuration of the simulation is implemented using the python environment. The models of JModelica.org are units of the FMI (Functional Mock-Up Interface) which is a tool independent standard to support both model exchange and co-simulation of dynamic models using a combination of xml-files and compiled C-code. The FMI approach relies on :

- a modelling environment describes a product sub-system by differential, algebraic and discrete equations with time, state and step-events. These models can be large for usage in off-line or on-line simulation or can be used in embedded control systems;

- such tools generate and export the component in an FMU (Functional Mock-up Unit);

- an FMU can then be imported in another environment to be executed;

- several FMUs can – by this way – cooperate at runtime through a co-simulation environment, thanks to the FMI definitions of their interfaces.

Python environment offers: First, the PyModelica package in order to compile the Modelica models into FMUs, the function used is *compile_fmu(modelica_models)*. Second, the PyFMI which is an interface for interacting with FMUs and enables for example loading, via the function *load_fmu(fmu_models)*, of FMU models, setting of model parameters and evaluation of model equations.
For Simulation FMUs, the solver is contained inside the FMU. A simulation is performed simply by using the *simulate* method. In the script in the appendix A, we simulate the model from its default starting time 0.0 to *simTime* using the specified simulation options *opts*. The result is returned and stored in *res*. The options for a simulation algorithm can be retrieved by calling the method *simulate_options*. Within the options, we can change the system equations solver, the discretization type, the integrator step size and the filtration of the desired model variables. The function *set(value, variable)* of the FMU model is called to initialize the state variables and the environment parameters.

Simulation Output
As said previously, the results of the simulation are stored in the variables *res*. This variable is an associative array where the name of the state variable and its values given by the simulator. This array can be plotted using the package pyplot of the python library matplotlib (see the Figure 5.5) or saved within external file (i.e: text file). The DMPPT models are compiled, loaded and simulated. The main class is Dmppt we can reach the overall models and variables of the CPS from this model. For example, the notation $res['Dmppt.plant.panel.I']$ is used to recover the value of the state variable I_{pan} given by the simulation process.

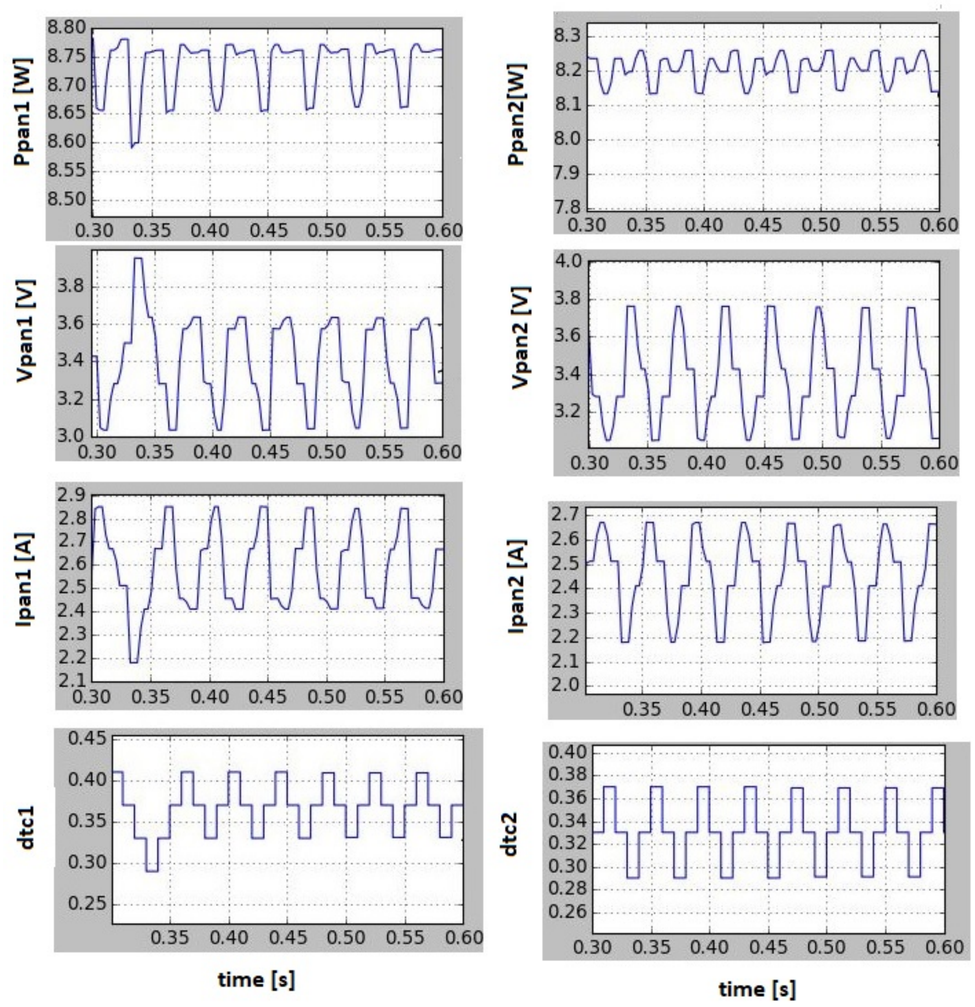
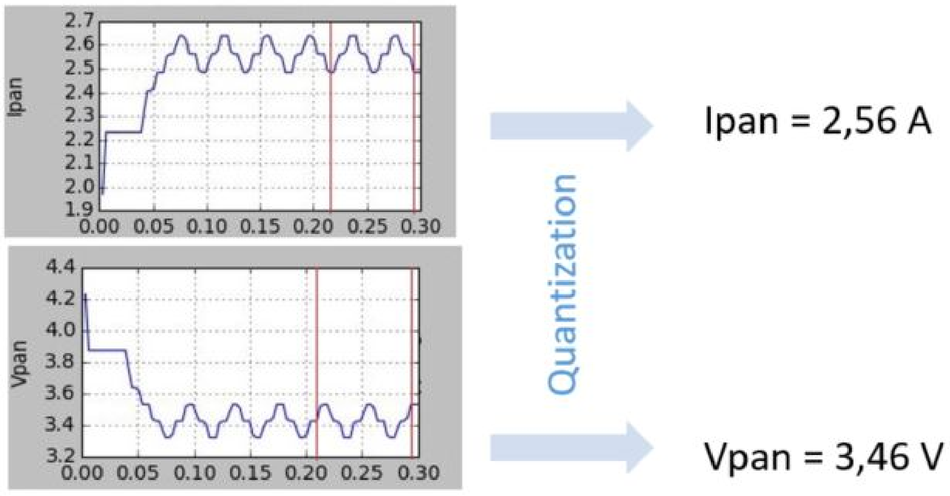


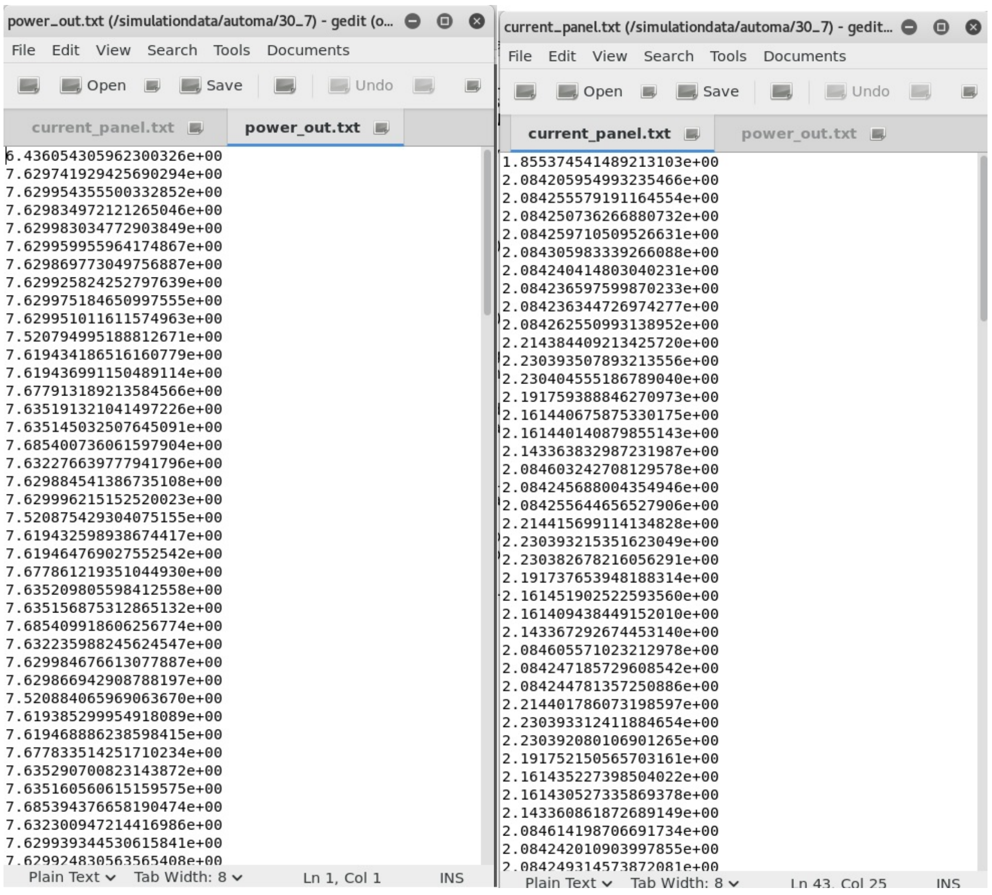
Figure 5.5 Plots of the simulation results using the pyplot package

The plot of the state variables is mandatory to the time-domain analysis of the steady state. By observing the plots, the expert can specify with accuracy the time period in which the system reaches the steady state. As instance, in the plot below the system clearly reaches the steady state during the time interval [0.21 s , 0.29 s]. At this point, the quantization function takes in input the values of the variables generated during the simulation (

$res['Dmppt.plant.panel.I']$ and $res['Dmppt.plant.panel.V']$), the starting time point $t_s = 0.21$ and the ending time point $t_f = 0.29$.



On the other hand, saving the values of the state variables on a external file (i.e : text file on the figure below) allows the frequency-domain analysis. By loading this files into an array, we can observe the frequency of the redundant values of every state variables easily, thus identifying the part of the array containing the values of the state variables when the system reaches the steady state.


power_out.txt (/simulationdata/automa/30_7) - gedit (o...
File Edit View Search Tools Documents
Open Save Undo
current_panel.txt
power_out.txt
Plain Text Tab Width: 8 Ln 1, Col 1 INS
current_panel.txt (/simulationdata/automa/30_7) - gedit...
File Edit View Search Tools Documents
Open Save Undo
current_panel.txt
power_out.txt
Plain Text Tab Width: 8 Ln 43, Col 25 INS

```
6.436054305962300326e+00
7.629741929425690294e+00
7.629954355500332852e+00
7.629834972121265046e+00
7.629983034772903849e+00
7.629959955964174867e+00
7.629869773049756887e+00
7.629925824252797639e+00
7.629975184650997555e+00
7.629951011611574963e+00
7.520794995188812671e+00
7.619434186516160779e+00
7.619436991150489114e+00
7.677913189213584566e+00
7.635191321041497226e+00
7.635145032507645091e+00
7.685400736061597904e+00
7.632276639777941796e+00
7.629884541386735108e+00
7.629996215152520023e+00
7.520875429304075155e+00
7.619432598938674417e+00
7.619464769027552542e+00
7.677861219351044930e+00
7.635209805598412558e+00
7.635156875312865132e+00
7.685409918606256774e+00
7.632235988245624547e+00
7.629984676613077887e+00
7.629866942908788197e+00
7.520884065969063670e+00
7.619385299954918089e+00
7.619468886238598415e+00
7.677833514251710234e+00
7.635290700823143872e+00
7.635160560615159575e+00
7.685394376658190474e+00
7.632300947214416986e+00
7.629939344530615841e+00
7.629924830563565408e+00
```

```
1.855374541489213103e+00
2.084205954993235466e+00
2.084255579191164554e+00
2.084250736266880732e+00
2.084259710509526631e+00
2.084305983339266088e+00
2.084240414803040231e+00
2.084236597599870233e+00
2.084236344726974277e+00
2.084262550993138952e+00
2.214384409213425720e+00
2.230393507893213556e+00
2.230404555186789040e+00
2.191759388846270973e+00
2.161440675875330175e+00
2.161440140879855143e+00
2.143363832987231987e+00
2.084603242708129578e+00
2.084245688004354946e+00
2.084255644656527906e+00
2.214415699114134828e+00
2.230393215351623049e+00
2.230382678216056291e+00
2.191737653948188314e+00
2.161451902522593560e+00
2.161409438449152010e+00
2.143367292674453140e+00
2.084605571023212978e+00
2.084247185729608542e+00
2.084244781357250886e+00
2.214401786073198597e+00
2.230393312411884654e+00
2.230392080106901265e+00
2.191752150565703161e+00
2.161435227398504022e+00
2.161430527335869378e+00
2.143360861872689149e+00
2.084614198706691734e+00
2.084242010903997855e+00
2.084249314573872081e+00
```

The overall methodology of the steady state analysis and the transition system definition for the DMPPT system are resumed into this 5 steps:

1. FOR ALL irradiation values, SIMULATE initializing with the quantized values of the state variables when the panels are not irradiated.

2. ADD the resulted states to the set of reached states

3. INITIALIZE the next simulations with the quantized outputs of the previous simulations and SIMULATE FOR ALL irradiation values.

4. CHECK if the resulted state is existing in the already reached states.
    - IF YES, then save the initialization state and the value of the irradiation. (transition)
    - IF NO, ADD the resulted state to the set of reached states, the initialization state and the value of the irradiation.
5. ITERATE 3 and 4 until no new state is reached.

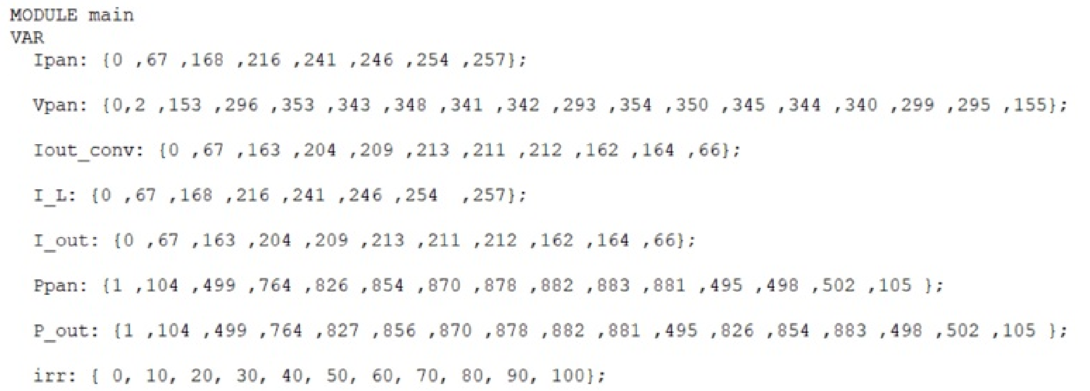
The final output of the overall simulation process is a finite transitions system described as following:

- The system's states are defined by the state variables: $\{I_{pan}, V_{pan}, I_L, I_{out-conv}, I_{out}, P_{pan}, P_{out}\}$.
- The input alphabet $\Sigma = \{$Nominal Irradiation percentage of the panels$\}$.
- The transitions between states are pre-computed during the simulation process.

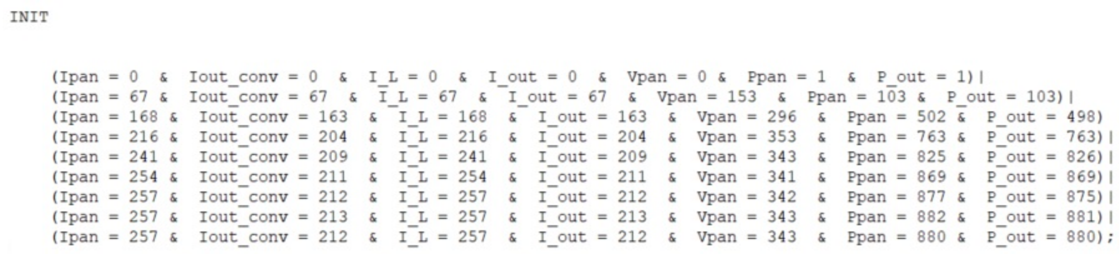
5.6 Model Checking : NuSMV

5.6.1 Defining the transition system in NuSMV description language

The input language of NuSMV is designed to allow for the description of Finite State Machines (FSM from now on) which range from completely synchronous to completely asynchronous, and from the detailed to the abstract. One can specify a system as a synchronous machine, or as an asynchronous network of non-deterministic processes. The language provides for modular hierarchical descriptions, and for the definition of reusable components. Since it is intended to describe finite state machines, the only data types in the language are finite ones – Integer, booleans, scalars and fixed arrays. Static data types can also be constructed. The primary purpose of the NuSMV input is to describe the transition relation of the FSM; this relation describes the valid evolution of the state of the FSM. In general, any propositional expression in the propositional calculus can be used to define the transition relation. This provides a great deal of flexibility, and at the same time a certain danger of inconsistency The simulations give us in output two sets: The set of system's states and the set of the transitions between those states. Based on this two sets, we generate a .smv file based on NuSMV description language where the system's states and the transitions are defined. First, we define the values of the state variables obtained during the simulation process. In the .smv file, the module main contains the values of the state variables : I_{pan}, V_{pan}, I_L, $I_{out-conv}$, I_{out}, P_{pan}, P_{out} and the admissible values of the panels irradiation irr. The space of states of the FSM is determined by the declarations of the state variables. The variables are declared to be of (predefined) scalar type, this means that it can assume the specified (integer) values, the NuSMV tool does not support the Real type, to this end a conversion from the real type given by JModelica.org simulator to the Integer type was crucial for the verification.


```
MODULE main
VAR
  Ipan: {0 ,67 ,168 ,216 ,241 ,246 ,254 ,257};

  Vpan: {0,2 ,153 ,296 ,353 ,343 ,348 ,341 ,342 ,293 ,354 ,350 ,345 ,344 ,340 ,299 ,295 ,155};

  Iout_conv: {0 ,67 ,163 ,204 ,209 ,213 ,211 ,212 ,162 ,164 ,66};

  I_L: {0 ,67 ,168 ,216 ,241 ,246 ,254  ,257};

  I_out: {0 ,67 ,163 ,204 ,209 ,213 ,211 ,212 ,162 ,164 ,66};

  Ppan: {1 ,104 ,499 ,764 ,826 ,854 ,870 ,878 ,882 ,883 ,881 ,495 ,498 ,502 ,105 };

  P_out: {1 ,104 ,499 ,764 ,827 ,856 ,870 ,878 ,882 ,881 ,495 ,826 ,854 ,883 ,498 ,502 ,105 };

  irr: { 0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100};
```

After that we specify the initial states within the block INIT the unpredictable behavior of the sun leads to the unknowledge of the initial value of the irradiation. Thus, the initial value of *irr* is completely unspecified, it can be one of the defined values which are chosen randomly. Because of this constraint, the initial state could eventually be one the admissible states, described later on, of the transition system.

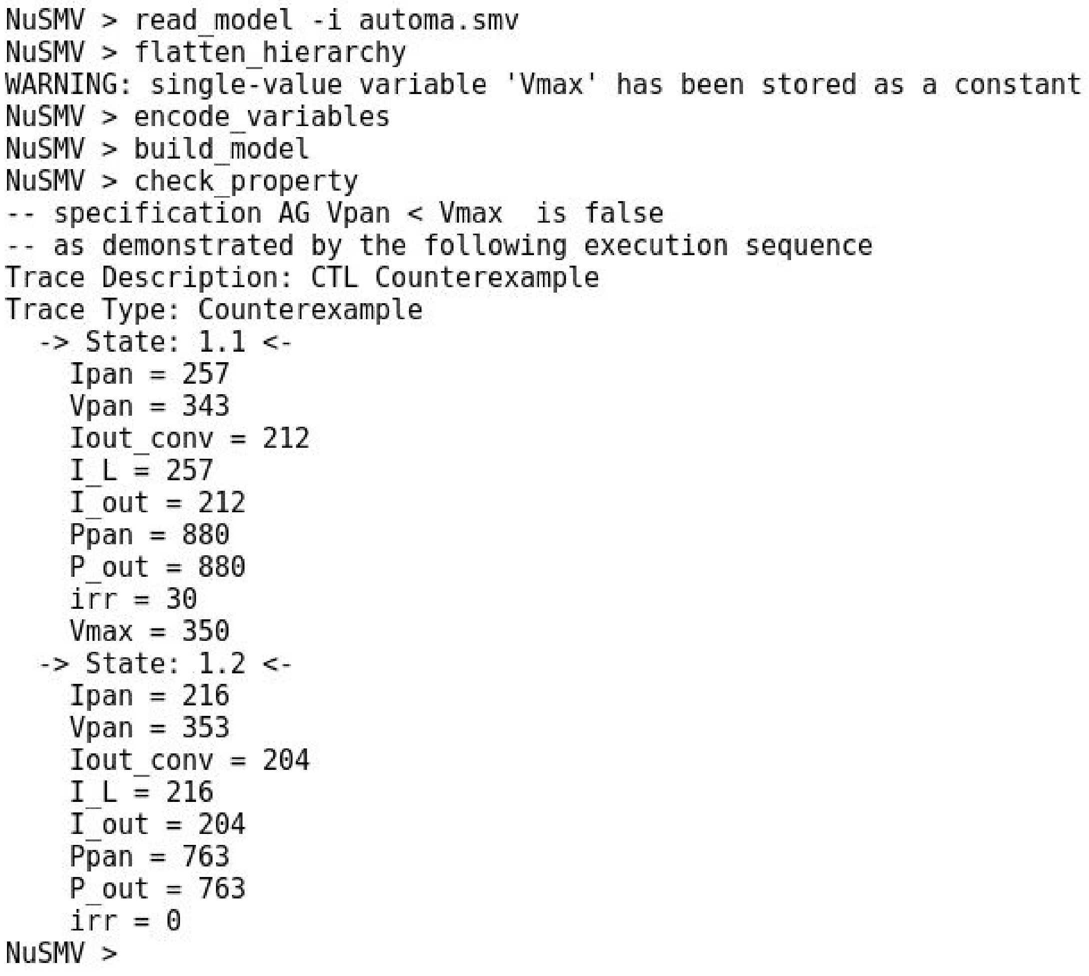


```
INIT


    (Ipan = 0  &  Iout_conv = 0  &  I_L = 0  &  I_out = 0  &  Vpan = 0 &  Ppan = 1  &  P_out = 1)|
    (Ipan = 67 &  Iout_conv = 67  &  I_L = 67  &  I_out = 67  &  Vpan = 153  &  Ppan = 103 &  P_out = 103)|
    (Ipan = 168 &  Iout_conv = 163  &  I_L = 168  &  I_out = 163  &  Vpan = 296  &  Ppan = 502 &  P_out = 498)
    (Ipan = 216 &  Iout_conv = 204  &  I_L = 216  &  I_out = 204  &  Vpan = 353  &  Ppan = 763 &  P_out = 763)|
    (Ipan = 241 &  Iout_conv = 209  &  I_L = 241  &  I_out = 209  &  Vpan = 343  &  Ppan = 825 &  P_out = 826)|
    (Ipan = 254 &  Iout_conv = 211  &  I_L = 254  &  I_out = 211  &  Vpan = 341  &  Ppan = 869 &  P_out = 869)|
    (Ipan = 257 &  Iout_conv = 212  &  I_L = 257  &  I_out = 212  &  Vpan = 342  &  Ppan = 877 &  P_out = 875)|
    (Ipan = 257 &  Iout_conv = 213  &  I_L = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 882 &  P_out = 881)|
    (Ipan = 257 &  Iout_conv = 212  &  I_L = 257  &  I_out = 212  &  Vpan = 343  &  Ppan = 880 &  P_out = 880);
```

The interpretation of the notation above is that the initial state is chosen randomly within the states declared in the block INIT. Those states are the representative values of the state variables resulted from the simulations : *for all irradiation values initializing with the quantized values of the state variables when the panels are not irradiated.*
The transition rules of the transition system is expressed by defining the value of variables in the next state (i.e. after each transition), given the value of variables in the current states (i.e. before the transition). With the ASSIGN block, the case segment sets the next value of each state variables to the next value depending on the irradiation and the previous values of the state variables (see the appendix B).

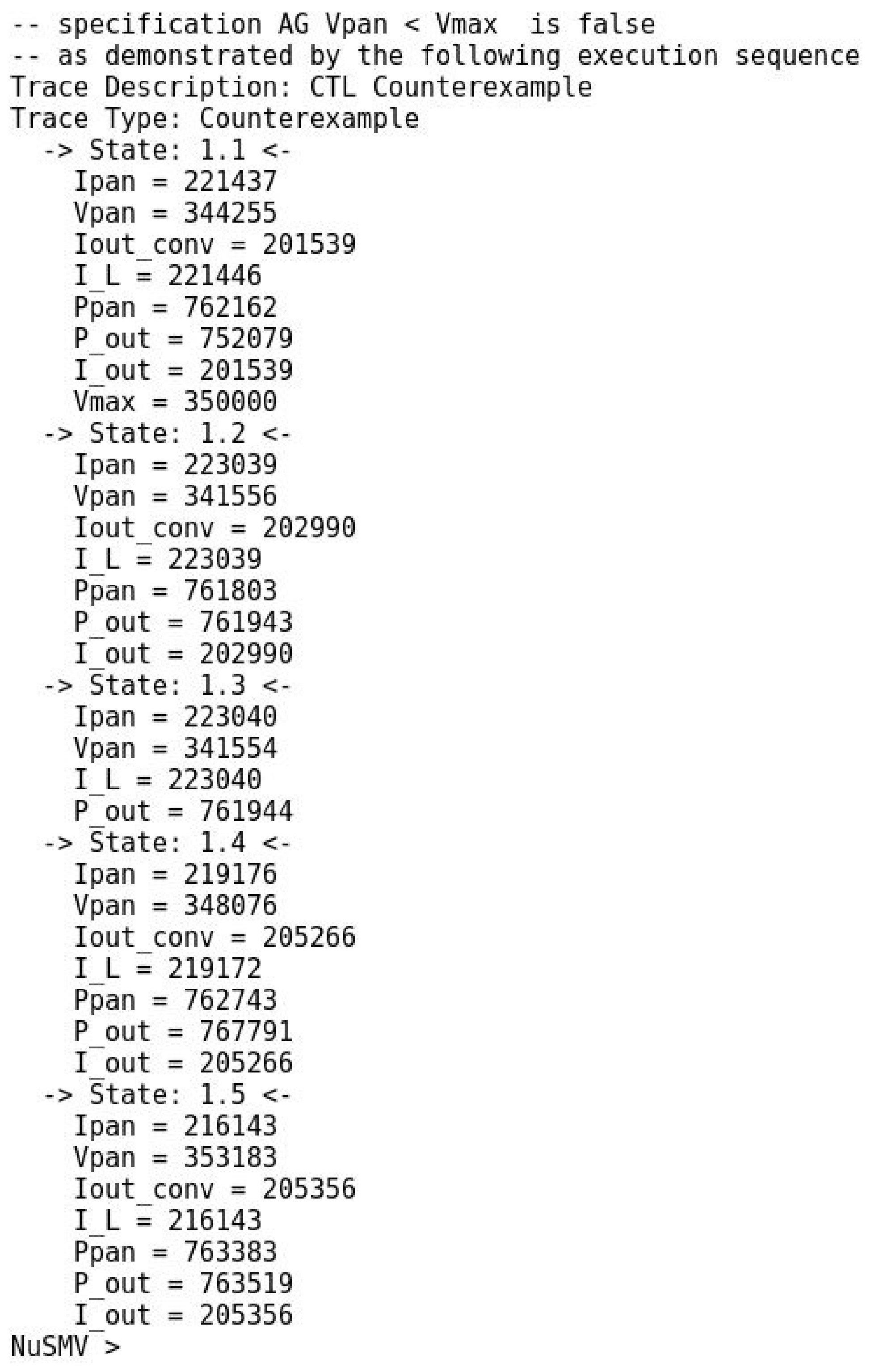
5.6.2 Verification of the transition system

At this step, the definition of the properties to verify can be performed. As said before, the properties are expressed in CTL or LTL. Let's take the example of the safety property "In every execution of the system, the value of the panel's voltage has to be lower than a maximum voltage threshold" . This can be expressed as a CTL property $\varphi = \mathbf{AG}\ (V_{pan} < V_{max})$. The results of the verification of this property is the following :



```
NuSMV > read_model -i automa.smv
NuSMV > flatten_hierarchy
WARNING: single-value variable 'Vmax' has been stored as a constant
NuSMV > encode_variables
NuSMV > build_model
NuSMV > check_property
-- specification AG Vpan < Vmax  is false
-- as demonstrated by the following execution sequence
Trace Description: CTL Counterexample
Trace Type: Counterexample
  -> State: 1.1 <-
    Ipan = 257
    Vpan = 343
    Iout_conv = 212
    I_L = 257
    I_out = 212
    Ppan = 880
    P_out = 880
    irr = 30
    Vmax = 350
  -> State: 1.2 <-
    Ipan = 216
    Vpan = 353
    Iout_conv = 204
    I_L = 216
    I_out = 204
    Ppan = 763
    P_out = 763
    irr = 0
NuSMV >
```

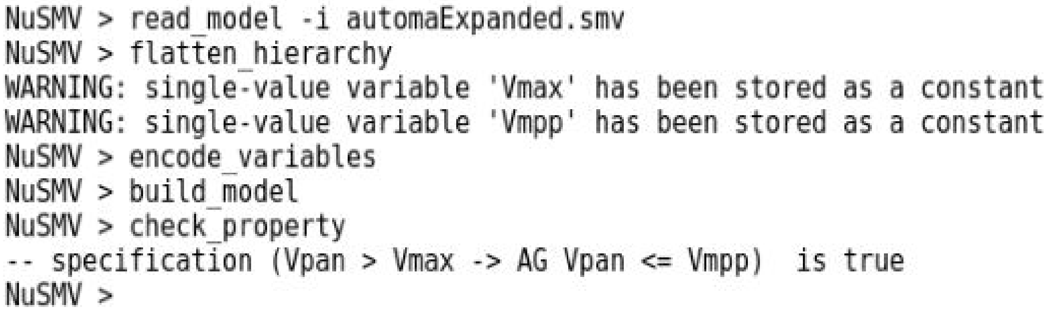
The property φ has been violated on the state S :$\{I_{pan} = 2.16$, $V_{pan} = 3.53$, $I_L = 2.16$, $I_{out-conv} = 2.04$, $I_{out} = 2.04$, $P_{pan} = 7.63$, $P_{out} = 7.63\}$. According to our verification methodology, we've traced back the state sequence S' resulted from the simulation which is represented by S and we've verified the same property on S'. The results of the verification of this property is the following :



```
-- specification AG Vpan < Vmax  is false
-- as demonstrated by the following execution sequence
Trace Description: CTL Counterexample
Trace Type: Counterexample
  -> State: 1.1 <-
    Ipan = 221437
    Vpan = 344255
    Iout_conv = 201539
    I_L = 221446
    Ppan = 762162
    P_out = 752079
    I_out = 201539
    Vmax = 350000
  -> State: 1.2 <-
    Ipan = 223039
    Vpan = 341556
    Iout_conv = 202990
    I_L = 223039
    Ppan = 761803
    P_out = 761943
    I_out = 202990
  -> State: 1.3 <-
    Ipan = 223040
    Vpan = 341554
    I_L = 223040
    P_out = 761944
  -> State: 1.4 <-
    Ipan = 219176
    Vpan = 348076
    Iout_conv = 205266
    I_L = 219172
    Ppan = 762743
    P_out = 767791
    I_out = 205266
  -> State: 1.5 <-
    Ipan = 216143
    Vpan = 353183
    Iout_conv = 205356
    I_L = 216143
    Ppan = 763383
    P_out = 763519
    I_out = 205356
NuSMV >
```

Within the state sequence, the property has been violated in the state $s' \in S'$ where s'=$\{I_{pan} = 2.16143$, $V_{pan} = 3.53183$, $I_L = 2.16143$, $I_{out-conv} = 2.05356$, $I_{out} = 2.05356$, $P_{pan} = 7.63383$, $P_{out} = 7.63519$ $\}$. The designer can determine exactly where and when the system became *faulty*.
At this point, we can check whether the violation of the *safety property* has an irreversible impact on the system by checking a *liveness* property which requires that the system is able to make progress despite the violation of the safety properties. Verifying that the system regulates its control in order to overcome this failure is an important aspect to investigate. To this end, we have checked, when the panel's voltage violates the property φ, if the system was able through the DMPPT mode controller to adjust the value of the voltage in such a way that the system comes out from the critical state. The liveness property we needed to verify is expressed in natural language in this way
"when V_{pan} is greater than the maximum voltage threshold in a state p of the system, check within all next states of p, if V_{pan} will always be less than or equal to the maximum power point voltage".

$$\psi = (V_{pan} > V_{max}) \rightarrow (\mathbf{AG}(V_{pan} <= V_{mpp}))$$



```
NuSMV > read_model -i automaExpanded.smv
NuSMV > flatten_hierarchy
WARNING: single-value variable 'Vmax' has been stored as a constant
WARNING: single-value variable 'Vmpp' has been stored as a constant
NuSMV > encode_variables
NuSMV > build_model
NuSMV > check_property
-- specification (Vpan > Vmax -> AG Vpan <= Vmpp)  is true
NuSMV >
```

The verification results show that ψ has been satisfied within the states sequence. The system was able to recover thanks to the mode control, but this does not exclude the fact that the erroneous safety property could have a major impact on the operation of the system, thus the intervention of the human expert is essential to guarantee the smooth performance of the system at hand, by investigating the cause of the system's failure and refining the system's components based on the error trace provided by the model checker.

5.7 Conclusion

In this chapter, we have presented the case study of photo-voltaic circuit where the CPS models are implemented with Modelica Language, the simulation of the models are performed on JModelica.org platform and the verification methodology was done using NuSMV symbolic model checker. Results show that our method can effectively verify the system at hand, by simulating it once, using the methodology and the tools presented previously. The last chapter draws the conclusions of this thesis and propose future research directions within this topic.

Chapter 6

Conclusion and Future Works

6.1 Conclusions

In this thesis, we have presented a Model Based Design methodology for minimizing the computational and time costs for the full automatic verification of a CPS. To achieve this we have adopted the following approaches.
The modeling of the CPS is done using the Modelica language acausal modeling approach that allows the re-usability, flexibility and modularity of the system's components.
The CPS is simulated once in order to extract the finite transition system composing the state space of the system under verification, the simulation are launched on the JModelica.org platform.
The steady state analysis is run in parallel with the simulations to calculate the representative state of the state sequences generated by a single or many simulations. The state space definition process handle the construction of the set containing the representative system's states and the transition rules of this set.
The generated transition system is given in input to the model checker tool NuSMV. The verification methodology consists first on verifying a property expressed in temporal logic within the transition system of the representative states. If the property is

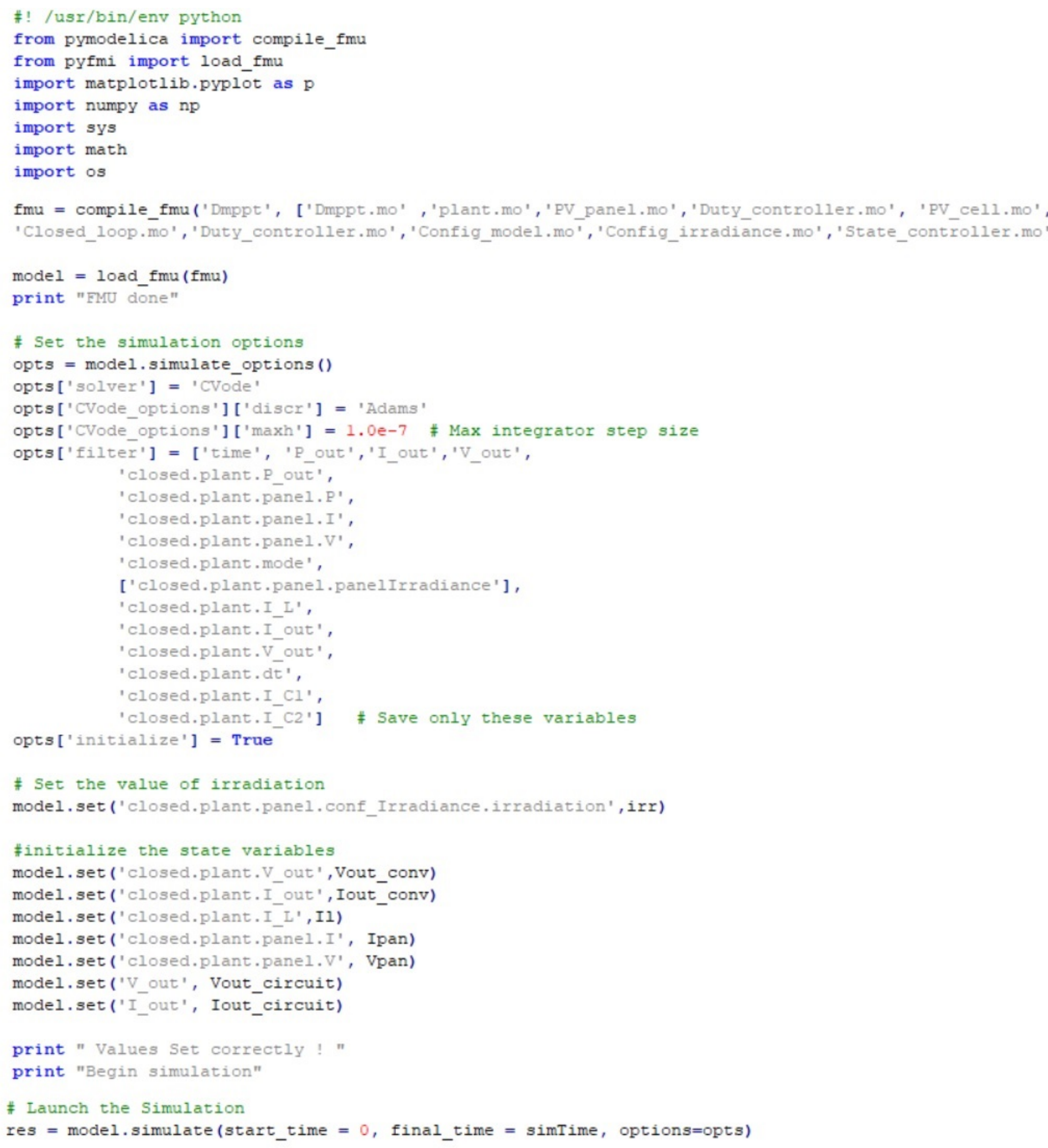
violated, the error is investigated by tracing-back the state sequences generated during the simulation process.
Finally, a case study of the DMPPT photo-voltaic system was presented among the verification results.

6.2 Future Works

As incoming works, we have planned to adopt a parallel execution of the simulations like the work presented in [80] to completely avoid the redundancy of the simulation scenarios that leads to the same representative state. Another enhancement could be done to the state space reduction technique to match steady-state responses as instance the partial order approaches [81] or model reduction procedure based on balanced state space representations [82]. The proposed approach could be applied to the verification of the systems that do not reach the steady state, by changing the way of choosing the representative state. In an extended version of this work, the temporal logic properties are expressed in graded modalities called also graded-CTL [83, 84, 85] which allows to count the paths leading to possible malfunctions, hence giving a grater potential of verification. An implementation of Graded-CTL in NuSMV is introduced with this work [86]. In [87], an approach was presented fitting perfectly for our finite transition system which is a Graded-CTL technique that adds a form of quantitative reasoning with no extra cost in computational complexity.

Appendix A

The Python script of the JModelica.rg simulation



```
#! /usr/bin/env python
from pymodelica import compile_fmu
from pyfmi import load_fmu
import matplotlib.pyplot as p
import numpy as np
import sys
import math
import os

fmu = compile_fmu('Dmppt', ['Dmppt.mo' ,'plant.mo','PV_panel.mo','Duty_controller.mo', 'PV_cell.mo',
'Closed_loop.mo','Duty_controller.mo','Config_model.mo','Config_irradiance.mo','State_controller.mo'

model = load_fmu(fmu)
print "FMU done"

# Set the simulation options
opts = model.simulate_options()
opts['solver'] = 'CVode'
opts['CVode_options']['discr'] = 'Adams'
opts['CVode_options']['maxh'] = 1.0e-7  # Max integrator step size
opts['filter'] = ['time', 'P_out','I_out','V_out',
          'closed.plant.P_out',
          'closed.plant.panel.P',
          'closed.plant.panel.I',
          'closed.plant.panel.V',
          'closed.plant.mode',
          ['closed.plant.panel.panelIrradiance'],
          'closed.plant.I_L',
          'closed.plant.I_out',
          'closed.plant.V_out',
          'closed.plant.dt',
          'closed.plant.I_C1',
          'closed.plant.I_C2']   # Save only these variables
opts['initialize'] = True

# Set the value of irradiation
model.set('closed.plant.panel.conf_Irradiance.irradiation',irr)

#initialize the state variables
model.set('closed.plant.V_out',Vout_conv)
model.set('closed.plant.I_out',Iout_conv)
model.set('closed.plant.I_L',Il)
model.set('closed.plant.panel.I', Ipan)
model.set('closed.plant.panel.V', Vpan)
model.set('V_out', Vout_circuit)
model.set('I_out', Iout_circuit)

print " Values Set correctly ! "
print "Begin simulation"

# Launch the Simulation
res = model.simulate(start_time = 0, final_time = simTime, options=opts)
```

Appendix B

The transition rules of the transition system

Within the block ASSIGN, the keyword **next** describes how the value of the variable changes in one step; again if the next value is unspecified, then the variable takes any value in its type at the next step (TRUE *variables_name*). The next value of each state variable depends on the previous values of the whole state variables and the irradiation value.

```
ASSIGN

  next(Ipan) := case
        Ipan = 0  &  Iout_conv = 0  &  I_L = 0  &  I_out = 0  &  Vpan = 0 &  Ppan = 1  &  P_out = 1 & irr = 0     : 1;
        Ipan = 0  &  Iout_conv = 0  &  I_L = 0  &  I_out = 0  &  Vpan = 0 &  Ppan = 1  &  P_out = 1 & irr = 10     : 67;
        Ipan = 0  &  Iout_conv = 0  &  I_L = 0  &  I_out = 0  &  Vpan = 0 &  Ppan = 1  &  P_out = 1 & irr = 20     : 168;
        Ipan = 0  &  Iout_conv = 0  &  I_L = 0  &  I_out = 0  &  Vpan = 0 &  Ppan = 1  &  P_out = 1 & irr = 30     : 217;
        Ipan = 0  &  Iout_conv = 0  &  I_L = 0  &  I_out = 0  &  Vpan = 0 &  Ppan = 1  &  P_out = 1 & irr = 40     : 241;
        Ipan = 0  &  Iout_conv = 0  &  I_L = 0  &  I_out = 0  &  Vpan = 0 &  Ppan = 1  &  P_out = 1 & irr = 50     : 247;
        Ipan = 0  &  Iout_conv = 0  &  I_L = 0  &  I_out = 0  &  Vpan = 0 &  Ppan = 1  &  P_out = 1 & irr = 60     : 255;
        Ipan = 0  &  Iout_conv = 0  &  I_L = 0  &  I_out = 0  &  Vpan = 0 &  Ppan = 1  &  P_out = 1 & irr = 70     : 257;
        Ipan = 0  &  Iout_conv = 0  &  I_L = 0  &  I_out = 0  &  Vpan = 0 &  Ppan = 1  &  P_out = 1 & irr = 80     : 257;
        Ipan = 0  &  Iout_conv = 0  &  I_L = 0  &  I_out = 0  &  Vpan = 0 &  Ppan = 1  &  P_out = 1 & irr = 90     : 257;
        Ipan = 0  &  Iout_conv = 0  &  I_L = 0  &  I_out = 0  &  Vpan = 0 &  Ppan = 1  &  P_out = 1 & irr = 100     : 257;
        Ipan = 67 &  Iout_conv = 67  &  I_L = 67  &  I_out = 67  &  Vpan = 153  &  Ppan = 103 &  P_out = 103 & irr = 0     : 0;
        Ipan = 67 &  Iout_conv = 67  &  I_L = 67  &  I_out = 67  &  Vpan = 153  &  Ppan = 103 &  P_out = 103 & irr = 10     : 67;
        Ipan = 67 &  Iout_conv = 67  &  I_L = 67  &  I_out = 67  &  Vpan = 153  &  Ppan = 103 &  P_out = 103 & irr = 20     : 168;
        Ipan = 67 &  Iout_conv = 67  &  I_L = 67  &  I_out = 67  &  Vpan = 153  &  Ppan = 103 &  P_out = 103 & irr = 30     : 216;
        Ipan = 67 &  Iout_conv = 67  &  I_L = 67  &  I_out = 67  &  Vpan = 153  &  Ppan = 103 &  P_out = 103 & irr = 40     : 241;
        Ipan = 67 &  Iout_conv = 67  &  I_L = 67  &  I_out = 67  &  Vpan = 153  &  Ppan = 103 &  P_out = 103 & irr = 50     : 246;
        Ipan = 67 &  Iout_conv = 67  &  I_L = 67  &  I_out = 67  &  Vpan = 153  &  Ppan = 103 &  P_out = 103 & irr = 60     : 254;
        Ipan = 67 &  Iout_conv = 67  &  I_L = 67  &  I_out = 67  &  Vpan = 153  &  Ppan = 103 &  P_out = 103 & irr = 70     : 257;
        Ipan = 67 &  Iout_conv = 67  &  I_L = 67  &  I_out = 67  &  Vpan = 153  &  Ppan = 103 &  P_out = 103 & irr = 80     : 257;
        Ipan = 67 &  Iout_conv = 67  &  I_L = 67  &  I_out = 67  &  Vpan = 153  &  Ppan = 103 &  P_out = 103 & irr = 90     : 257;
        Ipan = 67 &  Iout_conv = 67  &  I_L = 67  &  I_out = 67  &  Vpan = 153  &  Ppan = 103 &  P_out = 103 & irr = 100     : 257;
        Ipan = 168 &  Iout_conv = 163  &  I_L = 168  &  I_out = 163  &  Vpan = 296  &  Ppan = 498 &  P_out = 498 & irr = 0     : 0;
        Ipan = 168 &  Iout_conv = 163  &  I_L = 168  &  I_out = 163  &  Vpan = 296  &  Ppan = 498 &  P_out = 498 & irr = 10     : 67;
        Ipan = 168 &  Iout_conv = 163  &  I_L = 168  &  I_out = 163  &  Vpan = 296  &  Ppan = 498 &  P_out = 498 & irr = 20     : 168;
        Ipan = 168 &  Iout_conv = 163  &  I_L = 168  &  I_out = 163  &  Vpan = 296  &  Ppan = 498 &  P_out = 498 & irr = 30     : 216;
        Ipan = 168 &  Iout_conv = 163  &  I_L = 168  &  I_out = 163  &  Vpan = 296  &  Ppan = 498 &  P_out = 498 & irr = 40     : 241;
        Ipan = 168 &  Iout_conv = 163  &  I_L = 168  &  I_out = 163  &  Vpan = 296  &  Ppan = 498 &  P_out = 498 & irr = 50     : 246;
        Ipan = 168 &  Iout_conv = 163  &  I_L = 168  &  I_out = 163  &  Vpan = 296  &  Ppan = 498 &  P_out = 498 & irr = 60     : 254;
        Ipan = 168 &  Iout_conv = 163  &  I_L = 168  &  I_out = 163  &  Vpan = 296  &  Ppan = 498 &  P_out = 498 & irr = 70     : 254;
        Ipan = 168 &  Iout_conv = 163  &  I_L = 168  &  I_out = 163  &  Vpan = 296  &  Ppan = 498 &  P_out = 498 & irr = 80     : 257;
        Ipan = 168 &  Iout_conv = 163  &  I_L = 168  &  I_out = 163  &  Vpan = 296  &  Ppan = 498 &  P_out = 498 & irr = 90     : 257;
        Ipan = 168 &  Iout_conv = 163  &  I_L = 168  &  I_out = 163  &  Vpan = 296  &  Ppan = 498 &  P_out = 498 & irr = 100     : 257;
        Ipan = 216 &  Iout_conv = 204  &  I_L = 216  &  I_out = 204  &  Vpan = 353  &  Ppan = 763 &  P_out = 763 & irr = 0     : 0;
        Ipan = 216 &  Iout_conv = 204  &  I_L = 216  &  I_out = 204  &  Vpan = 353  &  Ppan = 763 &  P_out = 763 & irr = 10     : 67;
        Ipan = 216 &  Iout_conv = 204  &  I_L = 216  &  I_out = 204  &  Vpan = 353  &  Ppan = 763 &  P_out = 763 & irr = 20     : 168;
        Ipan = 216 &  Iout_conv = 204  &  I_L = 216  &  I_out = 204  &  Vpan = 353  &  Ppan = 763 &  P_out = 763 & irr = 30     : 216;
        Ipan = 216 &  Iout_conv = 204  &  I_L = 216  &  I_out = 204  &  Vpan = 353  &  Ppan = 763 &  P_out = 763 & irr = 40     : 241;
        Ipan = 216 &  Iout_conv = 204  &  I_L = 216  &  I_out = 204  &  Vpan = 353  &  Ppan = 763 &  P_out = 763 & irr = 50     : 246;
        Ipan = 216 &  Iout_conv = 204  &  I_L = 216  &  I_out = 204  &  Vpan = 353  &  Ppan = 763 &  P_out = 763 & irr = 60     : 257;
        Ipan = 216 &  Iout_conv = 204  &  I_L = 216  &  I_out = 204  &  Vpan = 353  &  Ppan = 763 &  P_out = 763 & irr = 70     : 254;
        Ipan = 216 &  Iout_conv = 204  &  I_L = 216  &  I_out = 204  &  Vpan = 353  &  Ppan = 763 &  P_out = 763 & irr = 80     : 257;
        Ipan = 216 &  Iout_conv = 204  &  I_L = 216  &  I_out = 204  &  Vpan = 353  &  Ppan = 763 &  P_out = 763 & irr = 90     : 257;
        Ipan = 216 &  Iout_conv = 204  &  I_L = 216  &  I_out = 204  &  Vpan = 353  &  Ppan = 763 &  P_out = 763 & irr = 100     : 257;
        Ipan = 241 &  Iout_conv = 209  &  I_L = 241  &  I_out = 209  &  Vpan = 343  &  Ppan = 825 &  P_out = 826 & irr = 0     : 0;
        Ipan = 241 &  Iout_conv = 209  &  I_L = 241  &  I_out = 209  &  Vpan = 343  &  Ppan = 825 &  P_out = 826 & irr = 10     : 67;
        Ipan = 241 &  Iout_conv = 209  &  I_L = 241  &  I_out = 209  &  Vpan = 343  &  Ppan = 825 &  P_out = 826 & irr = 20     : 168;
        Ipan = 241 &  Iout_conv = 209  &  I_L = 241  &  I_out = 209  &  Vpan = 343  &  Ppan = 825 &  P_out = 826 & irr = 30     : 216;
        Ipan = 241 &  Iout_conv = 209  &  I_L = 241  &  I_out = 209  &  Vpan = 343  &  Ppan = 825 &  P_out = 826 & irr = 40     : 241;
        Ipan = 241 &  Iout_conv = 209  &  I_L = 241  &  I_out = 209  &  Vpan = 343  &  Ppan = 825 &  P_out = 826 & irr = 50     : 246;
        Ipan = 241 &  Iout_conv = 209  &  I_L = 241  &  I_out = 209  &  Vpan = 343  &  Ppan = 825 &  P_out = 826 & irr = 60     : 254;
        Ipan = 241 &  Iout_conv = 209  &  I_L = 241  &  I_out = 209  &  Vpan = 343  &  Ppan = 825 &  P_out = 826 & irr = 70     : 257;
        Ipan = 241 &  Iout_conv = 209  &  I_L = 241  &  I_out = 209  &  Vpan = 343  &  Ppan = 825 &  P_out = 826 & irr = 80     : 257;
        Ipan = 241 &  Iout_conv = 209  &  I_L = 241  &  I_out = 209  &  Vpan = 343  &  Ppan = 825 &  P_out = 826 & irr = 90     : 257;
        Ipan = 241 &  Iout_conv = 209  &  I_L = 241  &  I_out = 209  &  Vpan = 343  &  Ppan = 825 &  P_out = 826 & irr = 100     : 257;
        Ipan = 246 &  Iout_conv = 213  &  I_L = 246  &  I_out = 213  &  Vpan = 348  &  Ppan = 853 &  P_out = 855 & irr = 0     : 0;
        Ipan = 246 &  Iout_conv = 213  &  I_L = 246  &  I_out = 213  &  Vpan = 348  &  Ppan = 853 &  P_out = 855 & irr = 10     : 67;
        Ipan = 246 &  Iout_conv = 213  &  I_L = 246  &  I_out = 213  &  Vpan = 348  &  Ppan = 853 &  P_out = 855 & irr = 20     : 168;
        Ipan = 246 &  Iout_conv = 213  &  I_L = 246  &  I_out = 213  &  Vpan = 348  &  Ppan = 853 &  P_out = 855 & irr = 30     : 216;
        Ipan = 246 &  Iout_conv = 213  &  I_L = 246  &  I_out = 213  &  Vpan = 348  &  Ppan = 853 &  P_out = 855 & irr = 40     : 241;
        Ipan = 246 &  Iout_conv = 213  &  I_L = 246  &  I_out = 213  &  Vpan = 348  &  Ppan = 853 &  P_out = 855 & irr = 50     : 246;
        Ipan = 246 &  Iout_conv = 213  &  I_L = 246  &  I_out = 213  &  Vpan = 348  &  Ppan = 853 &  P_out = 855 & irr = 60     : 254;
        Ipan = 246 &  Iout_conv = 213  &  I_L = 246  &  I_out = 213  &  Vpan = 348  &  Ppan = 853 &  P_out = 855 & irr = 70     : 257;
        Ipan = 246 &  Iout_conv = 213  &  I_L = 246  &  I_out = 213  &  Vpan = 348  &  Ppan = 853 &  P_out = 855 & irr = 80     : 257;
        Ipan = 246 &  Iout_conv = 213  &  I_L = 246  &  I_out = 213  &  Vpan = 348  &  Ppan = 853 &  P_out = 855 & irr = 90     : 257;
        Ipan = 246 &  Iout_conv = 213  &  I_L = 246  &  I_out = 213  &  Vpan = 348  &  Ppan = 853 &  P_out = 855 & irr = 100     : 257;
        Ipan = 254 &  Iout_conv = 211  &  I_L = 254  &  I_out = 211  &  Vpan = 341  &  Ppan = 869 &  P_out = 869 & irr = 0     : 0;
        Ipan = 254 &  Iout_conv = 211  &  I_L = 254  &  I_out = 211  &  Vpan = 341  &  Ppan = 869 &  P_out = 869 & irr = 10     : 67;
        Ipan = 254 &  Iout_conv = 211  &  I_L = 254  &  I_out = 211  &  Vpan = 341  &  Ppan = 869 &  P_out = 869 & irr = 20     : 168;
        Ipan = 254 &  Iout_conv = 211  &  I_L = 254  &  I_out = 211  &  Vpan = 341  &  Ppan = 869 &  P_out = 869 & irr = 30     : 216;
        Ipan = 254 &  Iout_conv = 211  &  I_L = 254  &  I_out = 211  &  Vpan = 341  &  Ppan = 869 &  P_out = 869 & irr = 40     : 241;
        Ipan = 254 &  Iout_conv = 211  &  I_L = 254  &  I_out = 211  &  Vpan = 341  &  Ppan = 869 &  P_out = 869 & irr = 50     : 246;
        Ipan = 254 &  Iout_conv = 211  &  I_L = 254  &  I_out = 211  &  Vpan = 341  &  Ppan = 869 &  P_out = 869 & irr = 60     : 254;
        Ipan = 254 &  Iout_conv = 211  &  I_L = 254  &  I_out = 211  &  Vpan = 341  &  Ppan = 869 &  P_out = 869 & irr = 70     : 254;
        Ipan = 254 &  Iout_conv = 211  &  I_L = 254  &  I_out = 211  &  Vpan = 341  &  Ppan = 869 &  P_out = 869 & irr = 80     : 257;
        Ipan = 254 &  Iout_conv = 211  &  I_L = 254  &  I_out = 211  &  Vpan = 341  &  Ppan = 869 &  P_out = 869 & irr = 90     : 257;
        Ipan = 254 &  Iout_conv = 211  &  I_L = 254  &  I_out = 211  &  Vpan = 341  &  Ppan = 869 &  P_out = 869 & irr = 100     : 257;
        Ipan = 257 &  Iout_conv = 212  &  I_L = 257  &  I_out = 212  &  Vpan = 342  &  Ppan = 877 &  P_out = 877 & irr = 0     : 0;
        Ipan = 257 &  Iout_conv = 212  &  I_L = 257  &  I_out = 212  &  Vpan = 342  &  Ppan = 877 &  P_out = 877 & irr = 10     : 67;
        Ipan = 257 &  Iout_conv = 212  &  I_L = 257  &  I_out = 212  &  Vpan = 342  &  Ppan = 877 &  P_out = 877 & irr = 20     : 168;
        Ipan = 257 &  Iout_conv = 212  &  I_L = 257  &  I_out = 212  &  Vpan = 342  &  Ppan = 877 &  P_out = 877 & irr = 30     : 216;
        Ipan = 257 &  Iout_conv = 212  &  I_L = 257  &  I_out = 212  &  Vpan = 342  &  Ppan = 877 &  P_out = 877 & irr = 40     : 241;
        Ipan = 257 &  Iout_conv = 212  &  I_L = 257  &  I_out = 212  &  Vpan = 342  &  Ppan = 877 &  P_out = 877 & irr = 50     : 246;
        Ipan = 257 &  Iout_conv = 212  &  I_L = 257  &  I_out = 212  &  Vpan = 342  &  Ppan = 877 &  P_out = 877 & irr = 60     : 254;
        Ipan = 257 &  Iout_conv = 212  &  I_L = 257  &  I_out = 212  &  Vpan = 342  &  Ppan = 877 &  P_out = 877 & irr = 70     : 257;
        Ipan = 257 &  Iout_conv = 212  &  I_L = 257  &  I_out = 212  &  Vpan = 342  &  Ppan = 877 &  P_out = 877 & irr = 80     : 257;
        Ipan = 257 &  Iout_conv = 212  &  I_L = 257  &  I_out = 212  &  Vpan = 342  &  Ppan = 877 &  P_out = 877 & irr = 90     : 257;
        Ipan = 257 &  Iout_conv = 212  &  I_L = 257  &  I_out = 212  &  Vpan = 342  &  Ppan = 877 &  P_out = 877 & irr = 100     : 257;
        Ipan = 257 &  Iout_conv = 213  &  I_L = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 881 &  P_out = 881 & irr = 0     : 0;
        Ipan = 257 &  Iout_conv = 213  &  I_L = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 881 &  P_out = 881 & irr = 10     : 67;
        Ipan = 257 &  Iout_conv = 213  &  I_L = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 881 &  P_out = 881 & irr = 20     : 168;
        Ipan = 257 &  Iout_conv = 213  &  I_L = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 881 &  P_out = 881 & irr = 30     : 216;
        Ipan = 257 &  Iout_conv = 213  &  I_L = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 881 &  P_out = 881 & irr = 40     : 241;
        Ipan = 257 &  Iout_conv = 213  &  I_L = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 881 &  P_out = 881 & irr = 50     : 246;
        Ipan = 257 &  Iout_conv = 213  &  I_L = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 881 &  P_out = 881 & irr = 60     : 254;
        Ipan = 257 &  Iout_conv = 213  &  I_L = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 881 &  P_out = 881 & irr = 70     : 257;
        Ipan = 257 &  Iout_conv = 213  &  I_L = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 881 &  P_out = 881 & irr = 80     : 257;
        Ipan = 257 &  Iout_conv = 213  &  I_L = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 881 &  P_out = 881 & irr = 90     : 257;
        Ipan = 257 &  Iout_conv = 213  &  I_L = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 881 &  P_out = 881 & irr = 100     : 257;
        Ipan = 257 &  Iout_conv = 213  &  I_L = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 882 &  P_out = 881 & irr = 0     : 0;
        Ipan = 257 &  Iout_conv = 213  &  I_L = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 882 &  P_out = 881 & irr = 10     : 67;
        Ipan = 257 &  Iout_conv = 213  &  I_L = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 882 &  P_out = 881 & irr = 20     : 168;
        Ipan = 257 &  Iout_conv = 213  &  I_L = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 882 &  P_out = 881 & irr = 30     : 216;
        Ipan = 257 &  Iout_conv = 213  &  I_L = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 882 &  P_out = 881 & irr = 40     : 241;
        Ipan = 257 &  Iout_conv = 213  &  I_L = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 882 &  P_out = 881 & irr = 50     : 246;
        Ipan = 257 &  Iout_conv = 213  &  I_L = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 882 &  P_out = 881 & irr = 60     : 254;
        Ipan = 257 &  Iout_conv = 213  &  I_L = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 882 &  P_out = 881 & irr = 70     : 257;
        Ipan = 257 &  Iout_conv = 213  &  I_L = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 882 &  P_out = 881 & irr = 80     : 257;
        Ipan = 257 &  Iout_conv = 213  &  I_L = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 882 &  P_out = 881 & irr = 90     : 257;
        Ipan = 257 &  Iout_conv = 213  &  I_L = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 882 &  P_out = 881 & irr = 100     : 257;
        Ipan = 257 &  Iout_conv = 212  &  I_L = 257  &  I_out = 212  &  Vpan = 343  &  Ppan = 880 &  P_out = 880 & irr = 0     : 0;
        Ipan = 257 &  Iout_conv = 212  &  I_L = 257  &  I_out = 212  &  Vpan = 343  &  Ppan = 880 &  P_out = 880 & irr = 10     : 67;
        Ipan = 257 &  Iout_conv = 212  &  I_L = 257  &  I_out = 212  &  Vpan = 343  &  Ppan = 880 &  P_out = 880 & irr = 20     : 168;
        Ipan = 257 &  Iout_conv = 212  &  I_L = 257  &  I_out = 212  &  Vpan = 343  &  Ppan = 880 &  P_out = 880 & irr = 30     : 216;
        Ipan = 257 &  Iout_conv = 212  &  I_L = 257  &  I_out = 212  &  Vpan = 343  &  Ppan = 880 &  P_out = 880 & irr = 40     : 241;
```

```
    Ipan = 257 &  Iout_conv = 212  &  I_L = 257  &  I_out = 212  &  Vpan = 343  &  Ppan = 880 &  P_out = 880 & irr = 50     : 246;
    Ipan = 257 &  Iout_conv = 212  &  I_L = 257  &  I_out = 212  &  Vpan = 343  &  Ppan = 880 &  P_out = 880 & irr = 60     : 254;
    Ipan = 257 &  Iout_conv = 212  &  I_L = 257  &  I_out = 212  &  Vpan = 343  &  Ppan = 880 &  P_out = 880 & irr = 70     : 257;
    Ipan = 257 &  Iout_conv = 212  &  I_L = 257  &  I_out = 212  &  Vpan = 343  &  Ppan = 880 &  P_out = 880 & irr = 80     : 257;
    Ipan = 257 &  Iout_conv = 212  &  I_L = 257  &  I_out = 212  &  Vpan = 343  &  Ppan = 880 &  P_out = 880 & irr = 90     : 257;
    Ipan = 257 &  Iout_conv = 212  &  I_L = 257  &  I_out = 212  &  Vpan = 343  &  Ppan = 880 &  P_out = 880 & irr = 100     : 257;
    TRUE                                              : Ipan;
esac;
next(I_L) := case
    I_L = 0  &  Iout_conv = 0  &  Ipan = 0  &  I_out = 0  &  Vpan = 0 &  Ppan = 1  &  P_out = 1  & irr = 0     : 0;
    I_L = 0  &  Iout_conv = 0  &  Ipan = 0  &  I_out = 0  &  Vpan = 0 &  Ppan = 1  &  P_out = 1  & irr = 10     : 67;
    I_L = 0  &  Iout_conv = 0  &  Ipan = 0  &  I_out = 0  &  Vpan = 0 &  Ppan = 1  &  P_out = 1  & irr = 20     : 168;
    I_L = 0  &  Iout_conv = 0  &  Ipan = 0  &  I_out = 0  &  Vpan = 0 &  Ppan = 1  &  P_out = 1  & irr = 30     : 216;
    I_L = 0  &  Iout_conv = 0  &  Ipan = 0  &  I_out = 0  &  Vpan = 0 &  Ppan = 1  &  P_out = 1  & irr = 40     : 241;
    I_L = 0  &  Iout_conv = 0  &  Ipan = 0  &  I_out = 0  &  Vpan = 0 &  Ppan = 1  &  P_out = 1  & irr = 50     : 246;
    I_L = 0  &  Iout_conv = 0  &  Ipan = 0  &  I_out = 0  &  Vpan = 0 &  Ppan = 1  &  P_out = 1  & irr = 60     : 254;
    I_L = 0  &  Iout_conv = 0  &  Ipan = 0  &  I_out = 0  &  Vpan = 0 &  Ppan = 1  &  P_out = 1  & irr = 70     : 257;
    I_L = 0  &  Iout_conv = 0  &  Ipan = 0  &  I_out = 0  &  Vpan = 0 &  Ppan = 1  &  P_out = 1  & irr = 80     : 257;
    I_L = 0  &  Iout_conv = 0  &  Ipan = 0  &  I_out = 0  &  Vpan = 0 &  Ppan = 1  &  P_out = 1  & irr = 90     : 257;
    I_L = 0  &  Iout_conv = 0  &  Ipan = 0  &  I_out = 0  &  Vpan = 0 &  Ppan = 1  &  P_out = 1  & irr = 100     : 257;
    I_L = 67 &  Iout_conv = 67  & Ipan = 67  &  I_out = 67  &  Vpan = 153  &  Ppan = 103 &  P_out = 103 & irr = 0     : 0;
    I_L = 67 &  Iout_conv = 67  & Ipan = 67  &  I_out = 67  &  Vpan = 153  &  Ppan = 103 &  P_out = 103 & irr = 10     : 67;
    I_L = 67 &  Iout_conv = 67  & Ipan = 67  &  I_out = 67  &  Vpan = 153  &  Ppan = 103 &  P_out = 103 & irr = 20     : 168;
    I_L = 67 &  Iout_conv = 67  & Ipan = 67  &  I_out = 67  &  Vpan = 153  &  Ppan = 103 &  P_out = 103 & irr = 30     : 216;
    I_L = 67 &  Iout_conv = 67  & Ipan = 67  &  I_out = 67  &  Vpan = 153  &  Ppan = 103 &  P_out = 103 & irr = 40     : 241;
    I_L = 67 &  Iout_conv = 67  & Ipan = 67  &  I_out = 67  &  Vpan = 153  &  Ppan = 103 &  P_out = 103 & irr = 50     : 246;
    I_L = 67 &  Iout_conv = 67  & Ipan = 67  &  I_out = 67  &  Vpan = 153  &  Ppan = 103 &  P_out = 103 & irr = 60     : 254;
    I_L = 67 &  Iout_conv = 67  & Ipan = 67  &  I_out = 67  &  Vpan = 153  &  Ppan = 103 &  P_out = 103 & irr = 70     : 257;
    I_L = 67 &  Iout_conv = 67  & Ipan = 67  &  I_out = 67  &  Vpan = 153  &  Ppan = 103 &  P_out = 103 & irr = 80     : 257;
    I_L = 67 &  Iout_conv = 67  & Ipan = 67  &  I_out = 67  &  Vpan = 153  &  Ppan = 103 &  P_out = 103 & irr = 90     : 257;
    I_L = 67 &  Iout_conv = 67  & Ipan = 67  &  I_out = 67  &  Vpan = 153  &  Ppan = 103 &  P_out = 103 & irr = 100     : 257;
    I_L = 168 &  Iout_conv = 163  & Ipan = 168  &  I_out = 163  &  Vpan = 296  &  Ppan = 498 &  P_out = 498 & irr = 0     : 0;
    I_L = 168 &  Iout_conv = 163  & Ipan = 168  &  I_out = 163  &  Vpan = 296  &  Ppan = 498 &  P_out = 498 & irr = 10     : 67;
    I_L = 168 &  Iout_conv = 163  & Ipan = 168  &  I_out = 163  &  Vpan = 296  &  Ppan = 498 &  P_out = 498 & irr = 20     : 168;
    I_L = 168 &  Iout_conv = 163  & Ipan = 168  &  I_out = 163  &  Vpan = 296  &  Ppan = 498 &  P_out = 498 & irr = 30     : 216;
    I_L = 168 &  Iout_conv = 163  & Ipan = 168  &  I_out = 163  &  Vpan = 296  &  Ppan = 498 &  P_out = 498 & irr = 40     : 241;
    I_L = 168 &  Iout_conv = 163  & Ipan = 168  &  I_out = 163  &  Vpan = 296  &  Ppan = 498 &  P_out = 498 & irr = 50     : 246;
    I_L = 168 &  Iout_conv = 163  & Ipan = 168  &  I_out = 163  &  Vpan = 296  &  Ppan = 498 &  P_out = 498 & irr = 60     : 254;
    I_L = 168 &  Iout_conv = 163  & Ipan = 168  &  I_out = 163  &  Vpan = 296  &  Ppan = 498 &  P_out = 498 & irr = 70     : 254;
    I_L = 168 &  Iout_conv = 163  & Ipan = 168  &  I_out = 163  &  Vpan = 296  &  Ppan = 498 &  P_out = 498 & irr = 80     : 257;
    I_L = 168 &  Iout_conv = 163  & Ipan = 168  &  I_out = 163  &  Vpan = 296  &  Ppan = 498 &  P_out = 498 & irr = 90     : 257;
    I_L = 168 &  Iout_conv = 163  & Ipan = 168  &  I_out = 163  &  Vpan = 296  &  Ppan = 498 &  P_out = 498 & irr = 100     : 257;
    I_L = 216 &  Iout_conv = 204  & Ipan = 216  &  I_out = 204  &  Vpan = 353  &  Ppan = 763 &  P_out = 763 & irr = 0     : 0;
    I_L = 216 &  Iout_conv = 204  & Ipan = 216  &  I_out = 204  &  Vpan = 353  &  Ppan = 763 &  P_out = 763 & irr = 10     : 67;
    I_L = 216 &  Iout_conv = 204  & Ipan = 216  &  I_out = 204  &  Vpan = 353  &  Ppan = 763 &  P_out = 763 & irr = 20     : 168;
    I_L = 216 &  Iout_conv = 204  & Ipan = 216  &  I_out = 204  &  Vpan = 353  &  Ppan = 763 &  P_out = 763 & irr = 30     : 216;
    I_L = 216 &  Iout_conv = 204  & Ipan = 216  &  I_out = 204  &  Vpan = 353  &  Ppan = 763 &  P_out = 763 & irr = 40     : 241;
    I_L = 216 &  Iout_conv = 204  & Ipan = 216  &  I_out = 204  &  Vpan = 353  &  Ppan = 763 &  P_out = 763 & irr = 50     : 246;
    I_L = 216 &  Iout_conv = 204  & Ipan = 216  &  I_out = 204  &  Vpan = 353  &  Ppan = 763 &  P_out = 763 & irr = 60     : 257;
    I_L = 216 &  Iout_conv = 204  & Ipan = 216  &  I_out = 204  &  Vpan = 353  &  Ppan = 763 &  P_out = 763 & irr = 70     : 254;
    I_L = 216 &  Iout_conv = 204  & Ipan = 216  &  I_out = 204  &  Vpan = 353  &  Ppan = 763 &  P_out = 763 & irr = 80     : 257;
    I_L = 216 &  Iout_conv = 204  & Ipan = 216  &  I_out = 204  &  Vpan = 353  &  Ppan = 763 &  P_out = 763 & irr = 90     : 257;
    I_L = 216 &  Iout_conv = 204  & Ipan = 216  &  I_out = 204  &  Vpan = 353  &  Ppan = 763 &  P_out = 763 & irr = 100     : 257;
    I_L = 241 &  Iout_conv = 209  & Ipan = 241  &  I_out = 209  &  Vpan = 343  &  Ppan = 825 &  P_out = 826 & irr = 0     : 0;
    I_L = 241 &  Iout_conv = 209  & Ipan = 241  &  I_out = 209  &  Vpan = 343  &  Ppan = 825 &  P_out = 826 & irr = 10     : 67;
    I_L = 241 &  Iout_conv = 209  & Ipan = 241  &  I_out = 209  &  Vpan = 343  &  Ppan = 825 &  P_out = 826 & irr = 20     : 168;
    I_L = 241 &  Iout_conv = 209  & Ipan = 241  &  I_out = 209  &  Vpan = 343  &  Ppan = 825 &  P_out = 826 & irr = 30     : 216;
    I_L = 241 &  Iout_conv = 209  & Ipan = 241  &  I_out = 209  &  Vpan = 343  &  Ppan = 825 &  P_out = 826 & irr = 40     : 241;
    I_L = 241 &  Iout_conv = 209  & Ipan = 241  &  I_out = 209  &  Vpan = 343  &  Ppan = 825 &  P_out = 826 & irr = 50     : 246;
    I_L = 241 &  Iout_conv = 209  & Ipan = 241  &  I_out = 209  &  Vpan = 343  &  Ppan = 825 &  P_out = 826 & irr = 60     : 254;
    I_L = 241 &  Iout_conv = 209  & Ipan = 241  &  I_out = 209  &  Vpan = 343  &  Ppan = 825 &  P_out = 826 & irr = 70     : 257;
    I_L = 241 &  Iout_conv = 209  & Ipan = 241  &  I_out = 209  &  Vpan = 343  &  Ppan = 825 &  P_out = 826 & irr = 80     : 257;
    I_L = 241 &  Iout_conv = 209  & Ipan = 241  &  I_out = 209  &  Vpan = 343  &  Ppan = 825 &  P_out = 826 & irr = 90     : 257;
    I_L = 241 &  Iout_conv = 209  & Ipan = 241  &  I_out = 209  &  Vpan = 343  &  Ppan = 825 &  P_out = 826 & irr = 100     : 257;
    I_L = 246 &  Iout_conv = 213  & Ipan = 246  &  I_out = 213  &  Vpan = 348  &  Ppan = 853 &  P_out = 855 & irr = 0     : 0;
    I_L = 246 &  Iout_conv = 213  & Ipan = 246  &  I_out = 213  &  Vpan = 348  &  Ppan = 853 &  P_out = 855 & irr = 10     : 67;
    I_L = 246 &  Iout_conv = 213  & Ipan = 246  &  I_out = 213  &  Vpan = 348  &  Ppan = 853 &  P_out = 855 & irr = 20     : 168;
    I_L = 246 &  Iout_conv = 213  & Ipan = 246  &  I_out = 213  &  Vpan = 348  &  Ppan = 853 &  P_out = 855 & irr = 30     : 216;
    I_L = 246 &  Iout_conv = 213  & Ipan = 246  &  I_out = 213  &  Vpan = 348  &  Ppan = 853 &  P_out = 855 & irr = 40     : 241;
    I_L = 246 &  Iout_conv = 213  & Ipan = 246  &  I_out = 213  &  Vpan = 348  &  Ppan = 853 &  P_out = 855 & irr = 50     : 246;
    I_L = 246 &  Iout_conv = 213  & Ipan = 246  &  I_out = 213  &  Vpan = 348  &  Ppan = 853 &  P_out = 855 & irr = 60     : 254;
    I_L = 246 &  Iout_conv = 213  & Ipan = 246  &  I_out = 213  &  Vpan = 348  &  Ppan = 853 &  P_out = 855 & irr = 70     : 257;
    I_L = 246 &  Iout_conv = 213  & Ipan = 246  &  I_out = 213  &  Vpan = 348  &  Ppan = 853 &  P_out = 855 & irr = 80     : 257;
    I_L = 246 &  Iout_conv = 213  & Ipan = 246  &  I_out = 213  &  Vpan = 348  &  Ppan = 853 &  P_out = 855 & irr = 90     : 257;
    I_L = 246 &  Iout_conv = 213  & Ipan = 246  &  I_out = 213  &  Vpan = 348  &  Ppan = 853 &  P_out = 855 & irr = 100     : 257;
    I_L = 254 &  Iout_conv = 211  & Ipan = 254  &  I_out = 211  &  Vpan = 341  &  Ppan = 869 &  P_out = 869 & irr = 0     : 0;
    I_L = 254 &  Iout_conv = 211  & Ipan = 254  &  I_out = 211  &  Vpan = 341  &  Ppan = 869 &  P_out = 869 & irr = 10     : 67;
    I_L = 254 &  Iout_conv = 211  & Ipan = 254  &  I_out = 211  &  Vpan = 341  &  Ppan = 869 &  P_out = 869 & irr = 20     : 168;
    I_L = 254 &  Iout_conv = 211  & Ipan = 254  &  I_out = 211  &  Vpan = 341  &  Ppan = 869 &  P_out = 869 & irr = 30     : 216;
    I_L = 254 &  Iout_conv = 211  & Ipan = 254  &  I_out = 211  &  Vpan = 341  &  Ppan = 869 &  P_out = 869 & irr = 40     : 241;
    I_L = 254 &  Iout_conv = 211  & Ipan = 254  &  I_out = 211  &  Vpan = 341  &  Ppan = 869 &  P_out = 869 & irr = 50     : 246;
    I_L = 254 &  Iout_conv = 211  & Ipan = 254  &  I_out = 211  &  Vpan = 341  &  Ppan = 869 &  P_out = 869 & irr = 60     : 254;
    I_L = 254 &  Iout_conv = 211  & Ipan = 254  &  I_out = 211  &  Vpan = 341  &  Ppan = 869 &  P_out = 869 & irr = 70     : 254;
    I_L = 254 &  Iout_conv = 211  & Ipan = 254  &  I_out = 211  &  Vpan = 341  &  Ppan = 869 &  P_out = 869 & irr = 80     : 257;
    I_L = 254 &  Iout_conv = 211  & Ipan = 254  &  I_out = 211  &  Vpan = 341  &  Ppan = 869 &  P_out = 869 & irr = 90     : 257;
    I_L = 254 &  Iout_conv = 211  & Ipan = 254  &  I_out = 211  &  Vpan = 341  &  Ppan = 869 &  P_out = 869 & irr = 100     : 257;
    I_L = 257 &  Iout_conv = 212  & Ipan = 257  &  I_out = 212  &  Vpan = 342  &  Ppan = 877 &  P_out = 877 & irr = 0     : 0;
    I_L = 257 &  Iout_conv = 212  & Ipan = 257  &  I_out = 212  &  Vpan = 342  &  Ppan = 877 &  P_out = 877 & irr = 10     : 67;
    I_L = 257 &  Iout_conv = 212  & Ipan = 257  &  I_out = 212  &  Vpan = 342  &  Ppan = 877 &  P_out = 877 & irr = 20     : 168;
    I_L = 257 &  Iout_conv = 212  & Ipan = 257  &  I_out = 212  &  Vpan = 342  &  Ppan = 877 &  P_out = 877 & irr = 30     : 216;
    I_L = 257 &  Iout_conv = 212  & Ipan = 257  &  I_out = 212  &  Vpan = 342  &  Ppan = 877 &  P_out = 877 & irr = 40     : 241;
    I_L = 257 &  Iout_conv = 212  & Ipan = 257  &  I_out = 212  &  Vpan = 342  &  Ppan = 877 &  P_out = 877 & irr = 50     : 246;
    I_L = 257 &  Iout_conv = 212  & Ipan = 257  &  I_out = 212  &  Vpan = 342  &  Ppan = 877 &  P_out = 877 & irr = 60     : 254;
    I_L = 257 &  Iout_conv = 212  & Ipan = 257  &  I_out = 212  &  Vpan = 342  &  Ppan = 877 &  P_out = 877 & irr = 70     : 257;
    I_L = 257 &  Iout_conv = 212  & Ipan = 257  &  I_out = 212  &  Vpan = 342  &  Ppan = 877 &  P_out = 877 & irr = 80     : 257;
    I_L = 257 &  Iout_conv = 212  & Ipan = 257  &  I_out = 212  &  Vpan = 342  &  Ppan = 877 &  P_out = 877 & irr = 90     : 257;
    I_L = 257 &  Iout_conv = 212  & Ipan = 257  &  I_out = 212  &  Vpan = 342  &  Ppan = 877 &  P_out = 877 & irr = 100     : 257;
    I_L = 257 &  Iout_conv = 213  & Ipan = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 881 &  P_out = 881 & irr = 0     : 0;
    I_L = 257 &  Iout_conv = 213  & Ipan = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 881 &  P_out = 881 & irr = 10     : 67;
    I_L = 257 &  Iout_conv = 213  & Ipan = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 881 &  P_out = 881 & irr = 20     : 168;
    I_L = 257 &  Iout_conv = 213  & Ipan = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 881 &  P_out = 881 & irr = 30     : 216;
    I_L = 257 &  Iout_conv = 213  & Ipan = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 881 &  P_out = 881 & irr = 40     : 241;
    I_L = 257 &  Iout_conv = 213  & Ipan = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 881 &  P_out = 881 & irr = 50     : 246;
    I_L = 257 &  Iout_conv = 213  & Ipan = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 881 &  P_out = 881 & irr = 60     : 254;
    I_L = 257 &  Iout_conv = 213  & Ipan = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 881 &  P_out = 881 & irr = 70     : 257;
    I_L = 257 &  Iout_conv = 213  & Ipan = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 881 &  P_out = 881 & irr = 80     : 257;
    I_L = 257 &  Iout_conv = 213  & Ipan = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 881 &  P_out = 881 & irr = 90     : 257;
    I_L = 257 &  Iout_conv = 213  & Ipan = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 881 &  P_out = 881 & irr = 100     : 257;
    I_L = 257 &  Iout_conv = 213  & Ipan = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 882 &  P_out = 881 & irr = 0     : 0;
    I_L = 257 &  Iout_conv = 213  & Ipan = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 882 &  P_out = 881 & irr = 10     : 67;
    I_L = 257 &  Iout_conv = 213  & Ipan = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 882 &  P_out = 881 & irr = 20     : 168;
    I_L = 257 &  Iout_conv = 213  & Ipan = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 882 &  P_out = 881 & irr = 30     : 216;
    I_L = 257 &  Iout_conv = 213  & Ipan = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 882 &  P_out = 881 & irr = 40     : 241;
    I_L = 257 &  Iout_conv = 213  & Ipan = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 882 &  P_out = 881 & irr = 50     : 246;
    I_L = 257 &  Iout_conv = 213  & Ipan = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 882 &  P_out = 881 & irr = 60     : 254;
    I_L = 257 &  Iout_conv = 213  & Ipan = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 882 &  P_out = 881 & irr = 70     : 257;
    I_L = 257 &  Iout_conv = 213  & Ipan = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 882 &  P_out = 881 & irr = 80     : 257;
    I_L = 257 &  Iout_conv = 213  & Ipan = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 882 &  P_out = 881 & irr = 90     : 257;
    I_L = 257 &  Iout_conv = 213  & Ipan = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 882 &  P_out = 881 & irr = 100     : 257;
    I_L = 257 &  Iout_conv = 212  & Ipan = 257  &  I_out = 212  &  Vpan = 343  &  Ppan = 880 &  P_out = 880 & irr = 0     : 0;
    I_L = 257 &  Iout_conv = 212  & Ipan = 257  &  I_out = 212  &  Vpan = 343  &  Ppan = 880 &  P_out = 880 & irr = 10     : 67;
    I_L = 257 &  Iout_conv = 212  & Ipan = 257  &  I_out = 212  &  Vpan = 343  &  Ppan = 880 &  P_out = 880 & irr = 20     : 168;
    I_L = 257 &  Iout_conv = 212  & Ipan = 257  &  I_out = 212  &  Vpan = 343  &  Ppan = 880 &  P_out = 880 & irr = 30     : 216;
    I_L = 257 &  Iout_conv = 212  & Ipan = 257  &  I_out = 212  &  Vpan = 343  &  Ppan = 880 &  P_out = 880 & irr = 40     : 241;
    I_L = 257 &  Iout_conv = 212  & Ipan = 257  &  I_out = 212  &  Vpan = 343  &  Ppan = 880 &  P_out = 880 & irr = 50     : 246;
    I_L = 257 &  Iout_conv = 212  & Ipan = 257  &  I_out = 212  &  Vpan = 343  &  Ppan = 880 &  P_out = 880 & irr = 60     : 254;
    I_L = 257 &  Iout_conv = 212  & Ipan = 257  &  I_out = 212  &  Vpan = 343  &  Ppan = 880 &  P_out = 880 & irr = 70     : 257;
    I_L = 257 &  Iout_conv = 212  & Ipan = 257  &  I_out = 212  &  Vpan = 343  &  Ppan = 880 &  P_out = 880 & irr = 80     : 257;
    I_L = 257 &  Iout_conv = 212  & Ipan = 257  &  I_out = 212  &  Vpan = 343  &  Ppan = 880 &  P_out = 880 & irr = 90     : 257;
    I_L = 257 &  Iout_conv = 212  & Ipan = 257  &  I_out = 212  &  Vpan = 343  &  Ppan = 880 &  P_out = 880 & irr = 100     : 257;
    TRUE                                              : I_L;
esac;
next(Iout_conv) := case
    Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &  I_out = 0  &  Vpan = 0 &  Ppan = 1  &  P_out = 1  & irr = 0     : 0;
    Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &  I_out = 0  &  Vpan = 0 &  Ppan = 1  &  P_out = 1  & irr = 10     : 67;
    Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &  I_out = 0  &  Vpan = 0 &  Ppan = 1  &  P_out = 1  & irr = 20     : 163;
    Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &  I_out = 0  &  Vpan = 0 &  Ppan = 1  &  P_out = 1  & irr = 30     : 204;
    Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &  I_out = 0  &  Vpan = 0 &  Ppan = 1  &  P_out = 1  & irr = 40     : 209;
    Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &  I_out = 0  &  Vpan = 0 &  Ppan = 1  &  P_out = 1  & irr = 50     : 213;
    Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &  I_out = 0  &  Vpan = 0 &  Ppan = 1  &  P_out = 1  & irr = 60     : 211;
    Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &  I_out = 0  &  Vpan = 0 &  Ppan = 1  &  P_out = 1  & irr = 70     : 212;
    Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &  I_out = 0  &  Vpan = 0 &  Ppan = 1  &  P_out = 1  & irr = 80     : 213;
    Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &  I_out = 0  &  Vpan = 0 &  Ppan = 1  &  P_out = 1  & irr = 90     : 213;
    Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &  I_out = 0  &  Vpan = 0 &  Ppan = 1  &  P_out = 1  & irr = 100     : 212;
    Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &  I_out = 67  &  Vpan = 153  &  Ppan = 103 &  P_out = 103 & irr = 0     : 0;
    Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &  I_out = 67  &  Vpan = 153  &  Ppan = 103 &  P_out = 103 & irr = 10     : 67;
    Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &  I_out = 67  &  Vpan = 153  &  Ppan = 103 &  P_out = 103 & irr = 20     : 162;
    Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &  I_out = 67  &  Vpan = 153  &  Ppan = 103 &  P_out = 103 & irr = 30     : 204;
    Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &  I_out = 67  &  Vpan = 153  &  Ppan = 103 &  P_out = 103 & irr = 40     : 209;
    Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &  I_out = 67  &  Vpan = 153  &  Ppan = 103 &  P_out = 103 & irr = 50     : 213;
    Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &  I_out = 67  &  Vpan = 153  &  Ppan = 103 &  P_out = 103 & irr = 60     : 211;
    Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &  I_out = 67  &  Vpan = 153  &  Ppan = 103 &  P_out = 103 & irr = 70     : 212;
    Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &  I_out = 67  &  Vpan = 153  &  Ppan = 103 &  P_out = 103 & irr = 80     : 213;
    Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &  I_out = 67  &  Vpan = 153  &  Ppan = 103 &  P_out = 103 & irr = 90     : 213;
    Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &  I_out = 67  &  Vpan = 153  &  Ppan = 103 &  P_out = 103 & irr = 100     : 212;
    Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &  I_out = 163  &  Vpan = 296  &  Ppan = 498 &  P_out = 498 & irr = 0     : 0;
    Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &  I_out = 163  &  Vpan = 296  &  Ppan = 498 &  P_out = 498 & irr = 10     : 67;
    Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &  I_out = 163  &  Vpan = 296  &  Ppan = 498 &  P_out = 498 & irr = 20     : 163;
    Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &  I_out = 163  &  Vpan = 296  &  Ppan = 498 &  P_out = 498 & irr = 30     : 204;
    Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &  I_out = 163  &  Vpan = 296  &  Ppan = 498 &  P_out = 498 & irr = 40     : 209;
    Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &  I_out = 163  &  Vpan = 296  &  Ppan = 498 &  P_out = 498 & irr = 50     : 213;
    Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &  I_out = 163  &  Vpan = 296  &  Ppan = 498 &  P_out = 498 & irr = 60     : 211;
    Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &  I_out = 163  &  Vpan = 296  &  Ppan = 498 &  P_out = 498 & irr = 70     : 213;
    Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &  I_out = 163  &  Vpan = 296  &  Ppan = 498 &  P_out = 498 & irr = 80     : 213;
    Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &  I_out = 163  &  Vpan = 296  &  Ppan = 498 &  P_out = 498 & irr = 90     : 213;
    Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &  I_out = 163  &  Vpan = 296  &  Ppan = 498 &  P_out = 498 & irr = 100     : 212;
    Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &  I_out = 204  &  Vpan = 353  &  Ppan = 763 &  P_out = 763 & irr = 0     : 0;
    Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &  I_out = 204  &  Vpan = 353  &  Ppan = 763 &  P_out = 763 & irr = 10     : 67;
    Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &  I_out = 204  &  Vpan = 353  &  Ppan = 763 &  P_out = 763 & irr = 20     : 162;
    Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &  I_out = 204  &  Vpan = 353  &  Ppan = 763 &  P_out = 763 & irr = 30     : 204;
    Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &  I_out = 204  &  Vpan = 353  &  Ppan = 763 &  P_out = 763 & irr = 40     : 209;
    Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &  I_out = 204  &  Vpan = 353  &  Ppan = 763 &  P_out = 763 & irr = 50     : 213;
    Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &  I_out = 204  &  Vpan = 353  &  Ppan = 763 &  P_out = 763 & irr = 60     : 211;
    Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &  I_out = 204  &  Vpan = 353  &  Ppan = 763 &  P_out = 763 & irr = 70     : 213;
    Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &  I_out = 204  &  Vpan = 353  &  Ppan = 763 &  P_out = 763 & irr = 80     : 213;
    Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &  I_out = 204  &  Vpan = 353  &  Ppan = 763 &  P_out = 763 & irr = 90     : 213;
    Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &  I_out = 204  &  Vpan = 353  &  Ppan = 763 &  P_out = 763 & irr = 100     : 213;
    Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &  I_out = 209  &  Vpan = 343  &  Ppan = 825 &  P_out = 826 & irr = 0     : 0;
    Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &  I_out = 209  &  Vpan = 343  &  Ppan = 825 &  P_out = 826 & irr = 10     : 67;
    Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &  I_out = 209  &  Vpan = 343  &  Ppan = 825 &  P_out = 826 & irr = 20     : 164;
    Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &  I_out = 209  &  Vpan = 343  &  Ppan = 825 &  P_out = 826 & irr = 30     : 204;
    Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &  I_out = 209  &  Vpan = 343  &  Ppan = 825 &  P_out = 826 & irr = 40     : 209;
    Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &  I_out = 209  &  Vpan = 343  &  Ppan = 825 &  P_out = 826 & irr = 50     : 213;
    Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &  I_out = 209  &  Vpan = 343  &  Ppan = 825 &  P_out = 826 & irr = 60     : 212;
    Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &  I_out = 209  &  Vpan = 343  &  Ppan = 825 &  P_out = 826 & irr = 70     : 212;
    Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &  I_out = 209  &  Vpan = 343  &  Ppan = 825 &  P_out = 826 & irr = 80     : 213;
    Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &  I_out = 209  &  Vpan = 343  &  Ppan = 825 &  P_out = 826 & irr = 90     : 213;
    Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &  I_out = 209  &  Vpan = 343  &  Ppan = 825 &  P_out = 826 & irr = 100     : 212;
    Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &  I_out = 213  &  Vpan = 348  &  Ppan = 853 &  P_out = 855 & irr = 0     : 0;
    Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &  I_out = 213  &  Vpan = 348  &  Ppan = 853 &  P_out = 855 & irr = 10     : 67;
    Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &  I_out = 213  &  Vpan = 348  &  Ppan = 853 &  P_out = 855 & irr = 20     : 162;
    Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &  I_out = 213  &  Vpan = 348  &  Ppan = 853 &  P_out = 855 & irr = 30     : 204;
    Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &  I_out = 213  &  Vpan = 348  &  Ppan = 853 &  P_out = 855 & irr = 40     : 209;
    Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &  I_out = 213  &  Vpan = 348  &  Ppan = 853 &  P_out = 855 & irr = 50     : 213;
    Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &  I_out = 213  &  Vpan = 348  &  Ppan = 853 &  P_out = 855 & irr = 60     : 212;
    Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &  I_out = 213  &  Vpan = 348  &  Ppan = 853 &  P_out = 855 & irr = 70     : 213;
    Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &  I_out = 213  &  Vpan = 348  &  Ppan = 853 &  P_out = 855 & irr = 80     : 213;
    Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &  I_out = 213  &  Vpan = 348  &  Ppan = 853 &  P_out = 855 & irr = 90     : 213;
    Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &  I_out = 213  &  Vpan = 348  &  Ppan = 853 &  P_out = 855 & irr = 100     : 212;
    Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &  I_out = 211  &  Vpan = 341  &  Ppan = 869 &  P_out = 869 & irr = 0     : 0;
    Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &  I_out = 211  &  Vpan = 341  &  Ppan = 869 &  P_out = 869 & irr = 10     : 67;
    Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &  I_out = 211  &  Vpan = 341  &  Ppan = 869 &  P_out = 869 & irr = 20     : 163;
    Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &  I_out = 211  &  Vpan = 341  &  Ppan = 869 &  P_out = 869 & irr = 30     : 204;
    Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &  I_out = 211  &  Vpan = 341  &  Ppan = 869 &  P_out = 869 & irr = 40     : 209;
    Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &  I_out = 211  &  Vpan = 341  &  Ppan = 869 &  P_out = 869 & irr = 50     : 213;
    Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &  I_out = 211  &  Vpan = 341  &  Ppan = 869 &  P_out = 869 & irr = 60     : 212;
    Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &  I_out = 211  &  Vpan = 341  &  Ppan = 869 &  P_out = 869 & irr = 70     : 213;
    Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &  I_out = 211  &  Vpan = 341  &  Ppan = 869 &  P_out = 869 & irr = 80     : 213;
    Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &  I_out = 211  &  Vpan = 341  &  Ppan = 869 &  P_out = 869 & irr = 90     : 213;
    Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &  I_out = 211  &  Vpan = 341  &  Ppan = 869 &  P_out = 869 & irr = 100     : 212;
    Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  I_out = 212  &  Vpan = 342  &  Ppan = 877 &  P_out = 877 & irr = 0     : 0;
    Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  I_out = 212  &  Vpan = 342  &  Ppan = 877 &  P_out = 877 & irr = 10     : 66;
    Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  I_out = 212  &  Vpan = 342  &  Ppan = 877 &  P_out = 877 & irr = 20     : 162;
    Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  I_out = 212  &  Vpan = 342  &  Ppan = 877 &  P_out = 877 & irr = 30     : 204;
    Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  I_out = 212  &  Vpan = 342  &  Ppan = 877 &  P_out = 877 & irr = 40     : 209;
    Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  I_out = 212  &  Vpan = 342  &  Ppan = 877 &  P_out = 877 & irr = 50     : 213;
    Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  I_out = 212  &  Vpan = 342  &  Ppan = 877 &  P_out = 877 & irr = 60     : 211;
    Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  I_out = 212  &  Vpan = 342  &  Ppan = 877 &  P_out = 877 & irr = 70     : 212;
    Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  I_out = 212  &  Vpan = 342  &  Ppan = 877 &  P_out = 877 & irr = 80     : 213;
    Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  I_out = 212  &  Vpan = 342  &  Ppan = 877 &  P_out = 877 & irr = 90     : 213;
    Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  I_out = 212  &  Vpan = 342  &  Ppan = 877 &  P_out = 877 & irr = 100     : 213;
    Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 881 &  P_out = 881 & irr = 0     : 0;
    Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 881 &  P_out = 881 & irr = 10     : 66;
    Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 881 &  P_out = 881 & irr = 20     : 162;
    Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 881 &  P_out = 881 & irr = 30     : 204;
    Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 881 &  P_out = 881 & irr = 40     : 209;
    Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 881 &  P_out = 881 & irr = 50     : 213;
    Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 881 &  P_out = 881 & irr = 60     : 211;
    Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 881 &  P_out = 881 & irr = 70     : 212;
    Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 881 &  P_out = 881 & irr = 80     : 213;
    Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 881 &  P_out = 881 & irr = 90     : 213;
    Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 881 &  P_out = 881 & irr = 100     : 213;
    Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 882 &  P_out = 881 & irr = 0     : 0;
    Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 882 &  P_out = 881 & irr = 10     : 66;
    Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 882 &  P_out = 881 & irr = 20     : 162;
    Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 882 &  P_out = 881 & irr = 30     : 204;
    Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 882 &  P_out = 881 & irr = 40     : 209;
    Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 882 &  P_out = 881 & irr = 50     : 213;
    Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 882 &  P_out = 881 & irr = 60     : 211;
    Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 882 &  P_out = 881 & irr = 70     : 212;
    Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 882 &  P_out = 881 & irr = 80     : 213;
    Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 882 &  P_out = 881 & irr = 90     : 213;
    Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  I_out = 213  &  Vpan = 343  &  Ppan = 882 &  P_out = 881 & irr = 100     : 213;
    Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  I_out = 212  &  Vpan = 343  &  Ppan = 880 &  P_out = 880 & irr = 0     : 0;
    Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  I_out = 212  &  Vpan = 343  &  Ppan = 880 &  P_out = 880 & irr = 10     : 66;
    Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  I_out = 212  &  Vpan = 343  &  Ppan = 880 &  P_out = 880 & irr = 20     : 162;
    Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  I_out = 212  &  Vpan = 343  &  Ppan = 880 &  P_out = 880 & irr = 30     : 204;
    Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  I_out = 212  &  Vpan = 343  &  Ppan = 880 &  P_out = 880 & irr = 40     : 209;
    Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  I_out = 212  &  Vpan = 343  &  Ppan = 880 &  P_out = 880 & irr = 50     : 213;
    Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  I_out = 212  &  Vpan = 343  &  Ppan = 880 &  P_out = 880 & irr = 60     : 211;
    Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  I_out = 212  &  Vpan = 343  &  Ppan = 880 &  P_out = 880 & irr = 70     : 212;
    Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  I_out = 212  &  Vpan = 343  &  Ppan = 880 &  P_out = 880 & irr = 80     : 213;
    Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  I_out = 212  &  Vpan = 343  &  Ppan = 880 &  P_out = 880 & irr = 90     : 213;
    Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  I_out = 212  &  Vpan = 343  &  Ppan = 880 &  P_out = 880 & irr = 100     : 213;
    TRUE                                          : Iout_conv;
esac;
next(I_out) := case
    I_out = 0   &  Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &   Vpan = 0  &  Ppan = 1  &  P_out = 1  & irr = 0     : 0;
    I_out = 0   &  Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &   Vpan = 0  &  Ppan = 1  &  P_out = 1  & irr = 10     : 67;
    I_out = 0   &  Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &   Vpan = 0  &  Ppan = 1  &  P_out = 1  & irr = 20     : 163;
    I_out = 0   &  Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &   Vpan = 0  &  Ppan = 1  &  P_out = 1  & irr = 30     : 204;
    I_out = 0   &  Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &   Vpan = 0  &  Ppan = 1  &  P_out = 1  & irr = 40     : 209;
    I_out = 0   &  Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &   Vpan = 0  &  Ppan = 1  &  P_out = 1  & irr = 50     : 213;
    I_out = 0   &  Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &   Vpan = 0  &  Ppan = 1  &  P_out = 1  & irr = 60     : 211;
    I_out = 0   &  Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &   Vpan = 0  &  Ppan = 1  &  P_out = 1  & irr = 70     : 212;
    I_out = 0   &  Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &   Vpan = 0  &  Ppan = 1  &  P_out = 1  & irr = 80     : 213;
    I_out = 0   &  Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &   Vpan = 0  &  Ppan = 1  &  P_out = 1  & irr = 90     : 213;
    I_out = 0   &  Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &   Vpan = 0  &  Ppan = 1  &  P_out = 1  & irr = 100     : 212;
    I_out = 67  &  Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &   Vpan = 153  &  Ppan = 103 &  P_out = 103 & irr = 0     : 0;
    I_out = 67  &  Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &   Vpan = 153  &  Ppan = 103 &  P_out = 103 & irr = 10     : 67;
    I_out = 67  &  Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &   Vpan = 153  &  Ppan = 103 &  P_out = 103 & irr = 20     : 162;
    I_out = 67  &  Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &   Vpan = 153  &  Ppan = 103 &  P_out = 103 & irr = 30     : 204;
    I_out = 67  &  Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &   Vpan = 153  &  Ppan = 103 &  P_out = 103 & irr = 40     : 209;
    I_out = 67  &  Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &   Vpan = 153  &  Ppan = 103 &  P_out = 103 & irr = 50     : 213;
    I_out = 67  &  Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &   Vpan = 153  &  Ppan = 103 &  P_out = 103 & irr = 60     : 211;
    I_out = 67  &  Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &   Vpan = 153  &  Ppan = 103 &  P_out = 103 & irr = 70     : 212;
    I_out = 67  &  Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &   Vpan = 153  &  Ppan = 103 &  P_out = 103 & irr = 80     : 213;
    I_out = 67  &  Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &   Vpan = 153  &  Ppan = 103 &  P_out = 103 & irr = 90     : 213;
    I_out = 67  &  Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &   Vpan = 153  &  Ppan = 103 &  P_out = 103 & irr = 100     : 212;
    I_out = 163  &  Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &   Vpan = 296  &  Ppan = 498 &  P_out = 498 & irr = 0     : 0;
    I_out = 163  &  Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &   Vpan = 296  &  Ppan = 498 &  P_out = 498 & irr = 10     : 67;
    I_out = 163  &  Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &   Vpan = 296  &  Ppan = 498 &  P_out = 498 & irr = 20     : 163;
    I_out = 163  &  Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &   Vpan = 296  &  Ppan = 498 &  P_out = 498 & irr = 30     : 204;
    I_out = 163  &  Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &   Vpan = 296  &  Ppan = 498 &  P_out = 498 & irr = 40     : 209;
    I_out = 163  &  Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &   Vpan = 296  &  Ppan = 498 &  P_out = 498 & irr = 50     : 213;
    I_out = 163  &  Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &   Vpan = 296  &  Ppan = 498 &  P_out = 498 & irr = 60     : 211;
    I_out = 163  &  Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &   Vpan = 296  &  Ppan = 498 &  P_out = 498 & irr = 70     : 213;
    I_out = 163  &  Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &   Vpan = 296  &  Ppan = 498 &  P_out = 498 & irr = 80     : 213;
    I_out = 163  &  Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &   Vpan = 296  &  Ppan = 498 &  P_out = 498 & irr = 90     : 213;
    I_out = 163  &  Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &   Vpan = 296  &  Ppan = 498 &  P_out = 498 & irr = 100     : 212;
    I_out = 204  &  Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &   Vpan = 353  &  Ppan = 763 &  P_out = 763 & irr = 0     : 0;
    I_out = 204  &  Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &   Vpan = 353  &  Ppan = 763 &  P_out = 763 & irr = 10     : 67;
    I_out = 204  &  Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &   Vpan = 353  &  Ppan = 763 &  P_out = 763 & irr = 20     : 162;
    I_out = 204  &  Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &   Vpan = 353  &  Ppan = 763 &  P_out = 763 & irr = 30     : 204;
    I_out = 204  &  Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &   Vpan = 353  &  Ppan = 763 &  P_out = 763 & irr = 40     : 209;
    I_out = 204  &  Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &   Vpan = 353  &  Ppan = 763 &  P_out = 763 & irr = 50     : 213;
    I_out = 204  &  Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &   Vpan = 353  &  Ppan = 763 &  P_out = 763 & irr = 60     : 211;
    I_out = 204  &  Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &   Vpan = 353  &  Ppan = 763 &  P_out = 763 & irr = 70     : 213;
    I_out = 204  &  Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &   Vpan = 353  &  Ppan = 763 &  P_out = 763 & irr = 80     : 213;
    I_out = 204  &  Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &   Vpan = 353  &  Ppan = 763 &  P_out = 763 & irr = 90     : 213;
    I_out = 204  &  Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &   Vpan = 353  &  Ppan = 763 &  P_out = 763 & irr = 100     : 213;
    I_out = 209  &  Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &   Vpan = 343  &  Ppan = 825 &  P_out = 826 & irr = 0     : 0;
    I_out = 209  &  Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &   Vpan = 343  &  Ppan = 825 &  P_out = 826 & irr = 10     : 67;
    I_out = 209  &  Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &   Vpan = 343  &  Ppan = 825 &  P_out = 826 & irr = 20     : 164;
    I_out = 209  &  Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &   Vpan = 343  &  Ppan = 825 &  P_out = 826 & irr = 30     : 204;
    I_out = 209  &  Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &   Vpan = 343  &  Ppan = 825 &  P_out = 826 & irr = 40     : 209;
    I_out = 209  &  Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &   Vpan = 343  &  Ppan = 825 &  P_out = 826 & irr = 50     : 213;
    I_out = 209  &  Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &   Vpan = 343  &  Ppan = 825 &  P_out = 826 & irr = 60     : 212;
    I_out = 209  &  Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &   Vpan = 343  &  Ppan = 825 &  P_out = 826 & irr = 70     : 212;
    I_out = 209  &  Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &   Vpan = 343  &  Ppan = 825 &  P_out = 826 & irr = 80     : 213;
    I_out = 209  &  Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &   Vpan = 343  &  Ppan = 825 &  P_out = 826 & irr = 90     : 213;
    I_out = 209  &  Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &   Vpan = 343  &  Ppan = 825 &  P_out = 826 & irr = 100     : 212;
    I_out = 213  &  Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &   Vpan = 348  &  Ppan = 853 &  P_out = 855 & irr = 0     : 0;
    I_out = 213  &  Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &   Vpan = 348  &  Ppan = 853 &  P_out = 855 & irr = 10     : 67;
    I_out = 213  &  Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &   Vpan = 348  &  Ppan = 853 &  P_out = 855 & irr = 20     : 162;
    I_out = 213  &  Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &   Vpan = 348  &  Ppan = 853 &  P_out = 855 & irr = 30     : 204;
    I_out = 213  &  Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &   Vpan = 348  &  Ppan = 853 &  P_out = 855 & irr = 40     : 209;
    I_out = 213  &  Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &   Vpan = 348  &  Ppan = 853 &  P_out = 855 & irr = 50     : 213;
    I_out = 213  &  Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &   Vpan = 348  &  Ppan = 853 &  P_out = 855 & irr = 60     : 212;
    I_out = 213  &  Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &   Vpan = 348  &  Ppan = 853 &  P_out = 855 & irr = 70     : 213;
    I_out = 213  &  Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &   Vpan = 348  &  Ppan = 853 &  P_out = 855 & irr = 80     : 213;
    I_out = 213  &  Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &   Vpan = 348  &  Ppan = 853 &  P_out = 855 & irr = 90     : 213;
    I_out = 213  &  Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &   Vpan = 348  &  Ppan = 853 &  P_out = 855 & irr = 100     : 212;
    I_out = 211  &  Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &   Vpan = 341  &  Ppan = 869 &  P_out = 869 & irr = 0     : 0;
    I_out = 211  &  Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &   Vpan = 341  &  Ppan = 869 &  P_out = 869 & irr = 10     : 67;
    I_out = 211  &  Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &   Vpan = 341  &  Ppan = 869 &  P_out = 869 & irr = 20     : 163;
    I_out = 211  &  Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &   Vpan = 341  &  Ppan = 869 &  P_out = 869 & irr = 30     : 204;
    I_out = 211  &  Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &   Vpan = 341  &  Ppan = 869 &  P_out = 869 & irr = 40     : 209;
    I_out = 211  &  Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &   Vpan = 341  &  Ppan = 869 &  P_out = 869 & irr = 50     : 213;
    I_out = 211  &  Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &   Vpan = 341  &  Ppan = 869 &  P_out = 869 & irr = 60     : 212;
    I_out = 211  &  Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &   Vpan = 341  &  Ppan = 869 &  P_out = 869 & irr = 70     : 213;
    I_out = 211  &  Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &   Vpan = 341  &  Ppan = 869 &  P_out = 869 & irr = 80     : 213;
    I_out = 211  &  Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &   Vpan = 341  &  Ppan = 869 &  P_out = 869 & irr = 90     : 213;
    I_out = 211  &  Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &   Vpan = 341  &  Ppan = 869 &  P_out = 869 & irr = 100     : 212;
    I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &   Vpan = 342  &  Ppan = 877 &  P_out = 877 & irr = 0     : 0;
    I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &   Vpan = 342  &  Ppan = 877 &  P_out = 877 & irr = 10     : 66;
    I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &   Vpan = 342  &  Ppan = 877 &  P_out = 877 & irr = 20     : 162;
    I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &   Vpan = 342  &  Ppan = 877 &  P_out = 877 & irr = 30     : 204;
    I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &   Vpan = 342  &  Ppan = 877 &  P_out = 877 & irr = 40     : 209;
    I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &   Vpan = 342  &  Ppan = 877 &  P_out = 877 & irr = 50     : 213;
    I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &   Vpan = 342  &  Ppan = 877 &  P_out = 877 & irr = 60     : 211;
    I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &   Vpan = 342  &  Ppan = 877 &  P_out = 877 & irr = 70     : 212;
    I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &   Vpan = 342  &  Ppan = 877 &  P_out = 877 & irr = 80     : 213;
    I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &   Vpan = 342  &  Ppan = 877 &  P_out = 877 & irr = 90     : 213;
    I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &   Vpan = 342  &  Ppan = 877 &  P_out = 877 & irr = 100     : 213;
    I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &   Vpan = 343  &  Ppan = 881 &  P_out = 881 & irr = 0     : 0;
    I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &   Vpan = 343  &  Ppan = 881 &  P_out = 881 & irr = 10     : 66;
    I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &   Vpan = 343  &  Ppan = 881 &  P_out = 881 & irr = 20     : 162;
    I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &   Vpan = 343  &  Ppan = 881 &  P_out = 881 & irr = 30     : 204;
    I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &   Vpan = 343  &  Ppan = 881 &  P_out = 881 & irr = 40     : 209;
    I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &   Vpan = 343  &  Ppan = 881 &  P_out = 881 & irr = 50     : 213;
    I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &   Vpan = 343  &  Ppan = 881 &  P_out = 881 & irr = 60     : 211;
    I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &   Vpan = 343  &  Ppan = 881 &  P_out = 881 & irr = 70     : 212;
    I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &   Vpan = 343  &  Ppan = 881 &  P_out = 881 & irr = 80     : 213;
    I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &   Vpan = 343  &  Ppan = 881 &  P_out = 881 & irr = 90     : 213;
    I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &   Vpan = 343  &  Ppan = 881 &  P_out = 881 & irr = 100     : 213;
    I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &   Vpan = 343  &  Ppan = 882 &  P_out = 881 & irr = 0     : 0;
    I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &   Vpan = 343  &  Ppan = 882 &  P_out = 881 & irr = 10     : 66;
    I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &   Vpan = 343  &  Ppan = 882 &  P_out = 881 & irr = 20     : 162;
    I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &   Vpan = 343  &  Ppan = 882 &  P_out = 881 & irr = 30     : 204;
    I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &    Vpan = 343  &  Ppan = 882 &  P_out = 881 & irr = 40     : 209;
    I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &    Vpan = 343  &  Ppan = 882 &  P_out = 881 & irr = 50     : 213;
    I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &    Vpan = 343  &  Ppan = 882 &  P_out = 881 & irr = 60     : 211;
    I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &    Vpan = 343  &  Ppan = 882 &  P_out = 881 & irr = 70     : 212;
    I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &    Vpan = 343  &  Ppan = 882 &  P_out = 881 & irr = 80     : 213;
    I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &    Vpan = 343  &  Ppan = 882 &  P_out = 881 & irr = 90     : 213;
    I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &    Vpan = 343  &  Ppan = 882 &  P_out = 881 & irr = 100     : 213;
    I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &    Vpan = 343  &  Ppan = 880 &  P_out = 880 & irr = 0     : 0;
    I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &    Vpan = 343  &  Ppan = 880 &  P_out = 880 & irr = 10     : 66;
    I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &    Vpan = 343  &  Ppan = 880 &  P_out = 880 & irr = 20     : 162;
    I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &    Vpan = 343  &  Ppan = 880 &  P_out = 880 & irr = 30     : 204;
    I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &    Vpan = 343  &  Ppan = 880 &  P_out = 880 & irr = 40     : 209;
    I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &    Vpan = 343  &  Ppan = 880 &  P_out = 880 & irr = 50     : 213;
    I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &    Vpan = 343  &  Ppan = 880 &  P_out = 880 & irr = 60     : 211;
    I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &    Vpan = 343  &  Ppan = 880 &  P_out = 880 & irr = 70     : 212;
    I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &    Vpan = 343  &  Ppan = 880 &  P_out = 880 & irr = 80     : 213;
    I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &    Vpan = 343  &  Ppan = 880 &  P_out = 880 & irr = 90     : 213;
    I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &    Vpan = 343  &  Ppan = 880 &  P_out = 880 & irr = 100     : 213;
    TRUE                                        : I_out;
esac;

next(Vpan) := case
    Vpan = 0   &  I_out = 0   &  Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &  Ppan = 1  &  P_out = 1  &  irr = 0     : 2;
    Vpan = 0   &  I_out = 0   &  Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &  Ppan = 1  &  P_out = 1  &  irr = 10     : 153;
    Vpan = 0   &  I_out = 0   &  Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &  Ppan = 1  &  P_out = 1  &  irr = 20     : 296;
    Vpan = 0   &  I_out = 0   &  Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &  Ppan = 1  &  P_out = 1  &  irr = 30     : 353;
    Vpan = 0   &  I_out = 0   &  Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &  Ppan = 1  &  P_out = 1  &  irr = 40     : 343;
    Vpan = 0   &  I_out = 0   &  Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &  Ppan = 1  &  P_out = 1  &  irr = 50     : 348;
    Vpan = 0   &  I_out = 0   &  Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &  Ppan = 1  &  P_out = 1  &  irr = 60     : 341;
    Vpan = 0   &  I_out = 0   &  Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &  Ppan = 1  &  P_out = 1  &  irr = 70     : 342;
    Vpan = 0   &  I_out = 0   &  Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &  Ppan = 1  &  P_out = 1  &  irr = 80     : 343;
    Vpan = 0   &  I_out = 0   &  Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &  Ppan = 1  &  P_out = 1  &  irr = 90     : 343;
    Vpan = 0   &  I_out = 0   &  Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &  Ppan = 1  &  P_out = 1  &  irr = 100     : 343;
    Vpan = 153  &   I_out = 67  &  Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &  Ppan = 103 &  P_out = 103  &  irr = 0     : 2;
    Vpan = 153  &   I_out = 67  &  Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &  Ppan = 103 &  P_out = 103  &  irr = 10     : 153;
    Vpan = 153  &   I_out = 67  &  Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &  Ppan = 103 &  P_out = 103  &  irr = 20     : 293;
    Vpan = 153  &   I_out = 67  &  Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &  Ppan = 103 &  P_out = 103  &  irr = 30     : 354;
    Vpan = 153  &   I_out = 67  &  Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &  Ppan = 103 &  P_out = 103  &  irr = 40     : 343;
    Vpan = 153  &   I_out = 67  &  Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &  Ppan = 103 &  P_out = 103  &  irr = 50     : 350;
    Vpan = 153  &   I_out = 67  &  Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &  Ppan = 103 &  P_out = 103  &  irr = 60     : 341;
    Vpan = 153  &   I_out = 67  &  Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &  Ppan = 103 &  P_out = 103  &  irr = 70     : 342;
    Vpan = 153  &   I_out = 67  &  Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &  Ppan = 103 &  P_out = 103  &  irr = 80     : 343;
    Vpan = 153  &   I_out = 67  &  Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &  Ppan = 103 &  P_out = 103  &  irr = 90     : 345;
    Vpan = 153  &   I_out = 67  &  Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &  Ppan = 103 &  P_out = 103  &  irr = 100     : 343;
    Vpan = 296  &   I_out = 163  &  Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &  Ppan = 498 &  P_out = 498  &  irr = 0     : 2;
    Vpan = 296  &   I_out = 163  &  Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &  Ppan = 498 &  P_out = 498  &  irr = 10     : 153;
    Vpan = 296  &   I_out = 163  &  Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &  Ppan = 498 &  P_out = 498  &  irr = 20     : 296;
    Vpan = 296  &   I_out = 163  &  Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &  Ppan = 498 &  P_out = 498  &  irr = 30     : 354;
    Vpan = 296  &   I_out = 163  &  Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &  Ppan = 498 &  P_out = 498  &  irr = 40     : 344;
    Vpan = 296  &   I_out = 163  &  Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &  Ppan = 498 &  P_out = 498  &  irr = 50     : 348;
    Vpan = 296  &   I_out = 163  &  Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &  Ppan = 498 &  P_out = 498  &  irr = 60     : 341;
    Vpan = 296  &   I_out = 163  &  Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &  Ppan = 498 &  P_out = 498  &  irr = 70     : 344;
    Vpan = 296  &   I_out = 163  &  Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &  Ppan = 498 &  P_out = 498  &  irr = 80     : 343;
    Vpan = 296  &   I_out = 163  &  Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &  Ppan = 498 &  P_out = 498  &  irr = 90     : 345;
    Vpan = 296  &   I_out = 163  &  Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &  Ppan = 498 &  P_out = 498  &  irr = 100     : 343;
    Vpan = 353  &   I_out = 204  &  Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &  Ppan = 763 &  P_out = 763  &  irr = 0     : 2;
    Vpan = 353  &   I_out = 204  &  Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &  Ppan = 763 &  P_out = 763  &  irr = 10     : 153;
    Vpan = 353  &   I_out = 204  &  Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &  Ppan = 763 &  P_out = 763  &  irr = 20     : 296;
    Vpan = 353  &   I_out = 204  &  Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &  Ppan = 763 &  P_out = 763  &  irr = 30     : 353;
    Vpan = 353  &   I_out = 204  &  Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &  Ppan = 763 &  P_out = 763  &  irr = 40     : 343;
    Vpan = 353  &   I_out = 204  &  Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &  Ppan = 763 &  P_out = 763  &  irr = 50     : 348;
    Vpan = 353  &   I_out = 204  &  Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &  Ppan = 763 &  P_out = 763  &  irr = 60     : 340;
    Vpan = 353  &   I_out = 204  &  Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &  Ppan = 763 &  P_out = 763  &  irr = 70     : 344;
    Vpan = 353  &   I_out = 204  &  Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &  Ppan = 763 &  P_out = 763  &  irr = 80     : 345;
    Vpan = 353  &   I_out = 204  &  Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &  Ppan = 763 &  P_out = 763  &  irr = 90     : 343;
    Vpan = 353  &   I_out = 204  &  Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &  Ppan = 763 &  P_out = 763  &  irr = 100     : 344;
    Vpan = 343  &   I_out = 209  &  Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &  Ppan = 825 &  P_out = 826  &  irr = 0     : 2;
    Vpan = 343  &   I_out = 209  &  Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &  Ppan = 825 &  P_out = 826  &  irr = 10     : 153;
    Vpan = 343  &   I_out = 209  &  Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &  Ppan = 825 &  P_out = 826  &  irr = 20     : 299;
    Vpan = 343  &   I_out = 209  &  Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &  Ppan = 825 &  P_out = 826  &  irr = 30     : 353;
    Vpan = 343  &   I_out = 209  &  Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &  Ppan = 825 &  P_out = 826  &  irr = 40     : 343;
    Vpan = 343  &   I_out = 209  &  Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &  Ppan = 825 &  P_out = 826  &  irr = 50     : 348;
    Vpan = 343  &   I_out = 209  &  Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &  Ppan = 825 &  P_out = 826  &  irr = 60     : 343;
    Vpan = 343  &   I_out = 209  &  Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &  Ppan = 825 &  P_out = 826  &  irr = 70     : 342;
    Vpan = 343  &   I_out = 209  &  Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &  Ppan = 825 &  P_out = 826  &  irr = 80     : 345;
    Vpan = 343  &   I_out = 209  &  Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &  Ppan = 825 &  P_out = 826  &  irr = 90     : 343;
    Vpan = 343  &   I_out = 209  &  Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &  Ppan = 825 &  P_out = 826  &  irr = 100     : 343;
    Vpan = 348  &   I_out = 213  &  Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &  Ppan = 853 &  P_out = 855  &  irr = 0     : 2;
    Vpan = 348  &   I_out = 213  &  Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &  Ppan = 853 &  P_out = 855  &  irr = 10     : 153;
    Vpan = 348  &   I_out = 213  &  Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &  Ppan = 853 &  P_out = 855  &  irr = 20     : 295;
    Vpan = 348  &   I_out = 213  &  Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &  Ppan = 853 &  P_out = 855  &  irr = 30     : 353;
    Vpan = 348  &   I_out = 213  &  Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &  Ppan = 853 &  P_out = 855  &  irr = 40     : 343;
    Vpan = 348  &   I_out = 213  &  Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &  Ppan = 853 &  P_out = 855  &  irr = 50     : 348;
    Vpan = 348  &   I_out = 213  &  Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &  Ppan = 853 &  P_out = 855  &  irr = 60     : 343;
    Vpan = 348  &   I_out = 213  &  Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &  Ppan = 853 &  P_out = 855  &  irr = 70     : 344;
    Vpan = 348  &   I_out = 213  &  Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &  Ppan = 853 &  P_out = 855  &  irr = 80     : 345;
    Vpan = 348  &   I_out = 213  &  Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &  Ppan = 853 &  P_out = 855  &  irr = 90     : 345;
    Vpan = 348  &   I_out = 213  &  Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &  Ppan = 853 &  P_out = 855  &  irr = 100     : 343;
    Vpan = 341  &   I_out = 211  &  Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &  Ppan = 869 &  P_out = 869  &  irr = 0     : 2;
    Vpan = 341  &   I_out = 211  &  Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &  Ppan = 869 &  P_out = 869  &  irr = 10     : 153;
    Vpan = 341  &   I_out = 211  &  Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &  Ppan = 869 &  P_out = 869  &  irr = 20     : 296;
    Vpan = 341  &   I_out = 211  &  Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &  Ppan = 869 &  P_out = 869  &  irr = 30     : 353;
    Vpan = 341  &   I_out = 211  &  Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &  Ppan = 869 &  P_out = 869  &  irr = 40     : 343;
    Vpan = 341  &   I_out = 211  &  Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &  Ppan = 869 &  P_out = 869  &  irr = 50     : 348;
    Vpan = 341  &   I_out = 211  &  Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &  Ppan = 869 &  P_out = 869  &  irr = 60     : 342;
    Vpan = 341  &   I_out = 211  &  Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &  Ppan = 869 &  P_out = 869  &  irr = 70     : 344;
    Vpan = 341  &   I_out = 211  &  Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &  Ppan = 869 &  P_out = 869  &  irr = 80     : 344;
    Vpan = 341  &   I_out = 211  &  Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &  Ppan = 869 &  P_out = 869  &  irr = 90     : 345;
    Vpan = 341  &   I_out = 211  &  Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &  Ppan = 869 &  P_out = 869  &  irr = 100     : 343;
    Vpan = 342  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Ppan = 877 &  P_out = 877  &  irr = 0     : 2;
    Vpan = 342  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Ppan = 877 &  P_out = 877  &  irr = 10     : 155;
    Vpan = 342  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Ppan = 877 &  P_out = 877  &  irr = 20     : 296;
    Vpan = 342  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Ppan = 877 &  P_out = 877  &  irr = 30     : 353;
    Vpan = 342  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Ppan = 877 &  P_out = 877  &  irr = 40     : 343;
    Vpan = 342  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Ppan = 877 &  P_out = 877  &  irr = 50     : 348;
    Vpan = 342  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Ppan = 877 &  P_out = 877  &  irr = 60     : 341;
    Vpan = 342  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Ppan = 877 &  P_out = 877  &  irr = 70     : 342;
    Vpan = 342  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Ppan = 877 &  P_out = 877  &  irr = 80     : 343;
    Vpan = 342  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Ppan = 877 &  P_out = 877  &  irr = 90     : 343;
    Vpan = 342  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Ppan = 877 &  P_out = 877  &  irr = 100     : 344;
    Vpan = 343  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Ppan = 881 &  P_out = 881  &  irr = 0     : 2;
    Vpan = 343  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Ppan = 881 &  P_out = 881  &  irr = 10     : 155;
    Vpan = 343  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Ppan = 881 &  P_out = 881  &  irr = 20     : 296;
    Vpan = 343  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Ppan = 881 &  P_out = 881  &  irr = 30     : 353;
    Vpan = 343  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Ppan = 881 &  P_out = 881  &  irr = 40     : 343;
    Vpan = 343  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Ppan = 881 &  P_out = 881  &  irr = 50     : 348;
    Vpan = 343  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Ppan = 881 &  P_out = 881  &  irr = 60     : 341;
    Vpan = 343  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Ppan = 881 &  P_out = 881  &  irr = 70     : 342;
    Vpan = 343  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Ppan = 881 &  P_out = 881  &  irr = 80     : 343;
    Vpan = 343  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Ppan = 881 &  P_out = 881  &  irr = 90     : 343;
    Vpan = 343  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Ppan = 881 &  P_out = 881  &  irr = 100     : 344;
    Vpan = 343  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Ppan = 882 &  P_out = 881  &  irr = 0     : 2;
    Vpan = 343  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Ppan = 882 &  P_out = 881  &  irr = 10     : 155;
    Vpan = 343  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Ppan = 882 &  P_out = 881  &  irr = 20     : 296;
    Vpan = 343  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Ppan = 882 &  P_out = 881  &  irr = 30     : 353;
    Vpan = 343  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Ppan = 882 &  P_out = 881  &  irr = 40     : 343;
    Vpan = 343  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Ppan = 882 &  P_out = 881  &  irr = 50     : 348;
    Vpan = 343  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Ppan = 882 &  P_out = 881  &  irr = 60     : 341;
    Vpan = 343  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Ppan = 882 &  P_out = 881  &  irr = 70     : 342;
    Vpan = 343  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Ppan = 882 &  P_out = 881  &  irr = 80     : 343;
    Vpan = 343  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Ppan = 882 &  P_out = 881  &  irr = 90     : 343;
    Vpan = 343  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Ppan = 882 &  P_out = 881  &  irr = 100     : 344;
    Vpan = 343  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Ppan = 880 &  P_out = 880  &  irr = 0     : 2;
    Vpan = 343  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Ppan = 880 &  P_out = 880  &  irr = 10     : 155;
    Vpan = 343  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Ppan = 880 &  P_out = 880  &  irr = 20     : 296;
    Vpan = 343  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Ppan = 880 &  P_out = 880  &  irr = 30     : 353;
    Vpan = 343  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Ppan = 880 &  P_out = 880  &  irr = 40     : 343;
    Vpan = 343  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Ppan = 880 &  P_out = 880  &  irr = 50     : 348;
    Vpan = 343  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Ppan = 880 &  P_out = 880  &  irr = 60     : 341;
    Vpan = 343  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Ppan = 880 &  P_out = 880  &  irr = 70     : 342;
    Vpan = 343  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Ppan = 880 &  P_out = 880  &  irr = 80     : 343;
    Vpan = 343  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Ppan = 880 &  P_out = 880  &  irr = 90     : 343;
    Vpan = 343  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Ppan = 880 &  P_out = 880  &  irr = 100     : 344;
    TRUE                                          : Vpan;
esac;
next(Ppan) := case
    Ppan = 1   &  I_out = 0   &  Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &  Vpan = 0  &  P_out = 1  &  irr = 0     : 0;
    Ppan = 1   &  I_out = 0   &  Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &  Vpan = 0  &  P_out = 1  &  irr = 10     : 103;
    Ppan = 1   &  I_out = 0   &  Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &  Vpan = 0  &  P_out = 1  &  irr = 20     : 498;
    Ppan = 1   &  I_out = 0   &  Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &  Vpan = 0  &  P_out = 1  &  irr = 30     : 763;
    Ppan = 1   &  I_out = 0   &  Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &  Vpan = 0  &  P_out = 1  &  irr = 40     : 825;
    Ppan = 1   &  I_out = 0   &  Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &  Vpan = 0  &  P_out = 1  &  irr = 50     : 853;
    Ppan = 1   &  I_out = 0   &  Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &  Vpan = 0  &  P_out = 1  &  irr = 60     : 869;
    Ppan = 1   &  I_out = 0   &  Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &  Vpan = 0  &  P_out = 1  &  irr = 70     : 877;
    Ppan = 1   &  I_out = 0   &  Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &  Vpan = 0  &  P_out = 1  &  irr = 80     : 881;
    Ppan = 1   &  I_out = 0   &  Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &  Vpan = 0  &  P_out = 1  &  irr = 90     : 882;
    Ppan = 1   &  I_out = 0   &  Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &  Vpan = 0  &  P_out = 1  &  irr = 100     : 880;
    Ppan = 103  &   I_out = 67  &  Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &  Vpan = 153 &  P_out = 103  &  irr = 0     : 0;
    Ppan = 103  &   I_out = 67  &  Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &  Vpan = 153 &  P_out = 103  &  irr = 10     : 103;
    Ppan = 103  &   I_out = 67  &  Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &  Vpan = 153 &  P_out = 103  &  irr = 20     : 494;
    Ppan = 103  &   I_out = 67  &  Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &  Vpan = 153 &  P_out = 103  &  irr = 30     : 763;
    Ppan = 103  &   I_out = 67  &  Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &  Vpan = 153 &  P_out = 103  &  irr = 40     : 825;
    Ppan = 103  &   I_out = 67  &  Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &  Vpan = 153 &  P_out = 103  &  irr = 50     : 853;
    Ppan = 103  &   I_out = 67  &  Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &  Vpan = 153 &  P_out = 103  &  irr = 60     : 869;
    Ppan = 103  &   I_out = 67  &  Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &  Vpan = 153 &  P_out = 103  &  irr = 70     : 877;
    Ppan = 103  &   I_out = 67  &  Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &  Vpan = 153 &  P_out = 103  &  irr = 80     : 881;
    Ppan = 103  &   I_out = 67  &  Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &  Vpan = 153 &  P_out = 103  &  irr = 90     : 882;
    Ppan = 103  &   I_out = 67  &  Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &  Vpan = 153 &  P_out = 103  &  irr = 100     : 880;
    Ppan = 498  &   I_out = 163  &  Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &  Vpan = 296 &  P_out = 498  &  irr = 0     : 0;
    Ppan = 498  &   I_out = 163  &  Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &  Vpan = 296 &  P_out = 498  &  irr = 10     : 103;
    Ppan = 498  &   I_out = 163  &  Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &  Vpan = 296 &  P_out = 498  &  irr = 20     : 498;
    Ppan = 498  &   I_out = 163  &  Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &  Vpan = 296 &  P_out = 498  &  irr = 30     : 763;
    Ppan = 498  &   I_out = 163  &  Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &  Vpan = 296 &  P_out = 498  &  irr = 40     : 825;
    Ppan = 498  &   I_out = 163  &  Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &  Vpan = 296 &  P_out = 498  &  irr = 50     : 853;
    Ppan = 498  &   I_out = 163  &  Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &  Vpan = 296 &  P_out = 498  &  irr = 60     : 869;
    Ppan = 498  &   I_out = 163  &  Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &  Vpan = 296 &  P_out = 498  &  irr = 70     : 877;
    Ppan = 498  &   I_out = 163  &  Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &  Vpan = 296 &  P_out = 498  &  irr = 80     : 881;
    Ppan = 498  &   I_out = 163  &  Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &  Vpan = 296 &  P_out = 498  &  irr = 90     : 882;
    Ppan = 498  &   I_out = 163  &  Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &  Vpan = 296 &  P_out = 498  &  irr = 100     : 880;
    Ppan = 763  &   I_out = 204  &  Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &  Vpan = 353 &  P_out = 763  &  irr = 0     : 0;
    Ppan = 763  &   I_out = 204  &  Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &  Vpan = 353 &  P_out = 763  &  irr = 10     : 103;
    Ppan = 763  &   I_out = 204  &  Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &  Vpan = 353 &  P_out = 763  &  irr = 20     : 497;
    Ppan = 763  &   I_out = 204  &  Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &  Vpan = 353 &  P_out = 763  &  irr = 30     : 763;
    Ppan = 763  &   I_out = 204  &  Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &  Vpan = 353 &  P_out = 763  &  irr = 40     : 825;
    Ppan = 763  &   I_out = 204  &  Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &  Vpan = 353 &  P_out = 763  &  irr = 50     : 853;
    Ppan = 763  &   I_out = 204  &  Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &  Vpan = 353 &  P_out = 763  &  irr = 60     : 869;
    Ppan = 763  &   I_out = 204  &  Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &  Vpan = 353 &  P_out = 763  &  irr = 70     : 877;
    Ppan = 763  &   I_out = 204  &  Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &  Vpan = 353 &  P_out = 763  &  irr = 80     : 881;
    Ppan = 763  &   I_out = 204  &  Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &  Vpan = 353 &  P_out = 763  &  irr = 90     : 882;
    Ppan = 763  &   I_out = 204  &  Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &  Vpan = 353 &  P_out = 763  &  irr = 100     : 880;
    Ppan = 825  &   I_out = 209  &  Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &  Vpan = 343 &  P_out = 826  &  irr = 0     : 0;
    Ppan = 825  &   I_out = 209  &  Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &  Vpan = 343 &  P_out = 826  &  irr = 10     : 103;
    Ppan = 825  &   I_out = 209  &  Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &  Vpan = 343 &  P_out = 826  &  irr = 20     : 501;
    Ppan = 825  &   I_out = 209  &  Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &  Vpan = 343 &  P_out = 826  &  irr = 30     : 763;
    Ppan = 825  &   I_out = 209  &  Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &  Vpan = 343 &  P_out = 826  &  irr = 40     : 825;
    Ppan = 825  &   I_out = 209  &  Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &  Vpan = 343 &  P_out = 826  &  irr = 50     : 853;
    Ppan = 825  &   I_out = 209  &  Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &  Vpan = 343 &  P_out = 826  &  irr = 60     : 869;
    Ppan = 825  &   I_out = 209  &  Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &  Vpan = 343 &  P_out = 826  &  irr = 70     : 877;
    Ppan = 825  &   I_out = 209  &  Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &  Vpan = 343 &  P_out = 826  &  irr = 80     : 881;
    Ppan = 825  &   I_out = 209  &  Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &  Vpan = 343 &  P_out = 826  &  irr = 90     : 882;
    Ppan = 825  &   I_out = 209  &  Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &  Vpan = 343 &  P_out = 826  &  irr = 100     : 880;
    Ppan = 853  &   I_out = 213  &  Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &  Vpan = 348 &  P_out = 855  &  irr = 0     : 0;
    Ppan = 853  &   I_out = 213  &  Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &  Vpan = 348 &  P_out = 855  &  irr = 10     : 103;
    Ppan = 853  &   I_out = 213  &  Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &  Vpan = 348 &  P_out = 855  &  irr = 20     : 497;
    Ppan = 853  &   I_out = 213  &  Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &  Vpan = 348 &  P_out = 855  &  irr = 30     : 763;
    Ppan = 853  &   I_out = 213  &  Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &  Vpan = 348 &  P_out = 855  &  irr = 40     : 825;
    Ppan = 853  &   I_out = 213  &  Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &  Vpan = 348 &  P_out = 855  &  irr = 50     : 853;
    Ppan = 853  &   I_out = 213  &  Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &  Vpan = 348 &  P_out = 855  &  irr = 60     : 869;
    Ppan = 853  &   I_out = 213  &  Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &  Vpan = 348 &  P_out = 855  &  irr = 70     : 877;
    Ppan = 853  &   I_out = 213  &  Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &  Vpan = 348 &  P_out = 855  &  irr = 80     : 881;
    Ppan = 853  &   I_out = 213  &  Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &  Vpan = 348 &  P_out = 855  &  irr = 90     : 882;
    Ppan = 853  &   I_out = 213  &  Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &  Vpan = 348 &  P_out = 855  &  irr = 100     : 880;
    Ppan = 869  &   I_out = 211  &  Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &  Vpan = 341 &  P_out = 869  &  irr = 0     : 0;
    Ppan = 869  &   I_out = 211  &  Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &  Vpan = 341 &  P_out = 869  &  irr = 10     : 103;
    Ppan = 869  &   I_out = 211  &  Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &  Vpan = 341 &  P_out = 869  &  irr = 20     : 498;
    Ppan = 869  &   I_out = 211  &  Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &  Vpan = 341 &  P_out = 869  &  irr = 30     : 763;
    Ppan = 869  &   I_out = 211  &  Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &  Vpan = 341 &  P_out = 869  &  irr = 40     : 825;
    Ppan = 869  &   I_out = 211  &  Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &  Vpan = 341 &  P_out = 869  &  irr = 50     : 853;
    Ppan = 869  &   I_out = 211  &  Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &  Vpan = 341 &  P_out = 869  &  irr = 60     : 869;
    Ppan = 869  &   I_out = 211  &  Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &  Vpan = 341 &  P_out = 869  &  irr = 70     : 877;
    Ppan = 869  &   I_out = 211  &  Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &  Vpan = 341 &  P_out = 869  &  irr = 80     : 881;
    Ppan = 869  &   I_out = 211  &  Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &  Vpan = 341 &  P_out = 869  &  irr = 90     : 882;
    Ppan = 869  &   I_out = 211  &  Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &  Vpan = 341 &  P_out = 869  &  irr = 100     : 880;
    Ppan = 877  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Vpan = 342 &  P_out = 877  &  irr = 0     : 0;
    Ppan = 877  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Vpan = 342 &  P_out = 877  &  irr = 10     : 104;
    Ppan = 877  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Vpan = 342 &  P_out = 877  &  irr = 20     : 497;
    Ppan = 877  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Vpan = 342 &  P_out = 877  &  irr = 30     : 763;
    Ppan = 877  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Vpan = 342 &  P_out = 877  &  irr = 40     : 825;
    Ppan = 877  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Vpan = 342 &  P_out = 877  &  irr = 50     : 853;
    Ppan = 877  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Vpan = 342 &  P_out = 877  &  irr = 60     : 869;
    Ppan = 877  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Vpan = 342 &  P_out = 877  &  irr = 70     : 877;
    Ppan = 877  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Vpan = 342 &  P_out = 877  &  irr = 80     : 881;
    Ppan = 877  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Vpan = 342 &  P_out = 877  &  irr = 90     : 882;
    Ppan = 877  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Vpan = 342 &  P_out = 877  &  irr = 100     : 880;
    Ppan = 881  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  P_out = 881  &  irr = 0     : 0;
    Ppan = 881  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  P_out = 881  &  irr = 10     : 104;
    Ppan = 881  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  P_out = 881  &  irr = 20     : 497;
    Ppan = 881  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  P_out = 881  &  irr = 30     : 763;
    Ppan = 881  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  P_out = 881  &  irr = 40     : 825;
```

```
        Ppan = 881  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  P_out = 881  &  irr = 50     : 853;
        Ppan = 881  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  P_out = 881  &  irr = 60     : 869;
        Ppan = 881  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  P_out = 881  &  irr = 70     : 877;
        Ppan = 881  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  P_out = 881  &  irr = 80     : 881;
        Ppan = 881  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  P_out = 881  &  irr = 90     : 882;
        Ppan = 881  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  P_out = 881  &  irr = 100     : 880;
        Ppan = 882  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  P_out = 881  &  irr = 0     : 0;
        Ppan = 882  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  P_out = 881  &  irr = 10     : 104;
        Ppan = 882  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  P_out = 881  &  irr = 20     : 497;
        Ppan = 882  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  P_out = 881  &  irr = 30     : 763;
        Ppan = 882  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  P_out = 881  &  irr = 40     : 825;
        Ppan = 882  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  P_out = 881  &  irr = 50     : 853;
        Ppan = 882  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  P_out = 881  &  irr = 60     : 869;
        Ppan = 882  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  P_out = 881  &  irr = 70     : 877;
        Ppan = 882  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  P_out = 881  &  irr = 80     : 881;
        Ppan = 882  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  P_out = 881  &  irr = 90     : 882;
        Ppan = 882  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  P_out = 881  &  irr = 100     : 880;
        Ppan = 880  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  P_out = 880  &  irr = 0     : 0;
        Ppan = 880  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  P_out = 880  &  irr = 10     : 104;
        Ppan = 880  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  P_out = 880  &  irr = 20     : 497;
        Ppan = 880  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  P_out = 880  &  irr = 30     : 763;
        Ppan = 880  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  P_out = 880  &  irr = 40     : 825;
        Ppan = 880  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  P_out = 880  &  irr = 50     : 853;
        Ppan = 880  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  P_out = 880  &  irr = 60     : 869;
        Ppan = 880  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  P_out = 880  &  irr = 70     : 877;
        Ppan = 880  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  P_out = 880  &  irr = 80     : 881;
        Ppan = 880  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  P_out = 880  &  irr = 90     : 882;
        Ppan = 880  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  P_out = 880  &  irr = 100     : 880;
        TRUE                                        : Ppan;
    esac;
    next(P_out) := case
        P_out = 1   &  I_out = 0   &  Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &  Ppan = 1  &  Vpan = 0  &  irr = 0     : 0;
        P_out = 1   &  I_out = 0   &  Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &  Ppan = 1  &  Vpan = 0  &  irr = 10     : 103;
        P_out = 1   &  I_out = 0   &  Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &  Ppan = 1  &  Vpan = 0  &  irr = 20     : 498;
        P_out = 1   &  I_out = 0   &  Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &  Ppan = 1  &  Vpan = 0  &  irr = 30     : 763;
        P_out = 1   &  I_out = 0   &  Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &  Ppan = 1  &  Vpan = 0  &  irr = 40     : 826;
        P_out = 1   &  I_out = 0   &  Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &  Ppan = 1  &  Vpan = 0  &  irr = 50     : 855;
        P_out = 1   &  I_out = 0   &  Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &  Ppan = 1  &  Vpan = 0  &  irr = 60     : 869;
        P_out = 1   &  I_out = 0   &  Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &  Ppan = 1  &  Vpan = 0  &  irr = 70     : 877;
        P_out = 1   &  I_out = 0   &  Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &  Ppan = 1  &  Vpan = 0  &  irr = 80     : 881;
        P_out = 1   &  I_out = 0   &  Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &  Ppan = 1  &  Vpan = 0  &  irr = 90     : 881;
        P_out = 1   &  I_out = 0   &  Iout_conv = 0  &  I_L = 0  &   Ipan = 0  &  Ppan = 1  &  Vpan = 0  &  irr = 100     : 880;
        P_out = 103  &   I_out = 67  &  Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &  Vpan = 153 &  Ppan = 103  &  irr = 0     : 0;
        P_out = 103  &   I_out = 67  &  Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &  Vpan = 153 &  Ppan = 103  &  irr = 10     : 103;
        P_out = 103  &   I_out = 67  &  Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &  Vpan = 153 &  Ppan = 103  &  irr = 20     : 494;
        P_out = 103  &   I_out = 67  &  Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &  Vpan = 153 &  Ppan = 103  &  irr = 30     : 763;
        P_out = 103  &   I_out = 67  &  Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &  Vpan = 153 &  Ppan = 103  &  irr = 40     : 825;
        P_out = 103  &   I_out = 67  &  Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &  Vpan = 153 &  Ppan = 103  &  irr = 50     : 853;
        P_out = 103  &   I_out = 67  &  Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &  Vpan = 153 &  Ppan = 103  &  irr = 60     : 869;
        P_out = 103  &   I_out = 67  &  Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &  Vpan = 153 &  Ppan = 103  &  irr = 70     : 877;
        P_out = 103  &   I_out = 67  &  Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &  Vpan = 153 &  Ppan = 103  &  irr = 80     : 881;
        P_out = 103  &   I_out = 67  &  Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &  Vpan = 153 &  Ppan = 103  &  irr = 90     : 882;
        P_out = 103  &   I_out = 67  &  Iout_conv = 67  &  I_L = 67 &  Ipan = 67  &  Vpan = 153 &  Ppan = 103  &  irr = 100     : 880;
        P_out = 498  &   I_out = 163  &  Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &  Vpan = 296 &  Ppan = 498  &  irr = 0     : 0;
        P_out = 498  &   I_out = 163  &  Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &  Vpan = 296 &  Ppan = 498  &  irr = 10     : 103;
        P_out = 498  &   I_out = 163  &  Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &  Vpan = 296 &  Ppan = 498  &  irr = 20     : 498;
        P_out = 498  &   I_out = 163  &  Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &  Vpan = 296 &  Ppan = 498  &  irr = 30     : 763;
        P_out = 498  &   I_out = 163  &  Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &  Vpan = 296 &  Ppan = 498  &  irr = 40     : 826;
        P_out = 498  &   I_out = 163  &  Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &  Vpan = 296 &  Ppan = 498  &  irr = 50     : 855;
        P_out = 498  &   I_out = 163  &  Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &  Vpan = 296 &  Ppan = 498  &  irr = 60     : 869;
        P_out = 498  &   I_out = 163  &  Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &  Vpan = 296 &  Ppan = 498  &  irr = 70     : 877;
        P_out = 498  &   I_out = 163  &  Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &  Vpan = 296 &  Ppan = 498  &  irr = 80     : 881;
        P_out = 498  &   I_out = 163  &  Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &  Vpan = 296 &  Ppan = 498  &  irr = 90     : 882;
        P_out = 498  &   I_out = 163  &  Iout_conv = 163  &  I_L = 168 &  Ipan = 168  &  Vpan = 296 &  Ppan = 498  &  irr = 100     : 880;
        P_out = 763  &   I_out = 204  &  Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &  Vpan = 353 &  Ppan = 763  &  irr = 0     : 0;
        P_out = 763  &   I_out = 204  &  Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &  Vpan = 353 &  Ppan = 763  &  irr = 10     : 103;
        P_out = 763  &   I_out = 204  &  Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &  Vpan = 353 &  Ppan = 763  &  irr = 20     : 497;
        P_out = 763  &   I_out = 204  &  Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &  Vpan = 353 &  Ppan = 763  &  irr = 30     : 763;
        P_out = 763  &   I_out = 204  &  Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &  Vpan = 353 &  Ppan = 763  &  irr = 40     : 825;
        P_out = 763  &   I_out = 204  &  Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &  Vpan = 353 &  Ppan = 763  &  irr = 50     : 855;
        P_out = 763  &   I_out = 204  &  Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &  Vpan = 353 &  Ppan = 763  &  irr = 60     : 869;
        P_out = 763  &   I_out = 204  &  Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &  Vpan = 353 &  Ppan = 763  &  irr = 70     : 877;
        P_out = 763  &   I_out = 204  &  Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &  Vpan = 353 &  Ppan = 763  &  irr = 80     : 881;
        P_out = 763  &   I_out = 204  &  Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &  Vpan = 353 &  Ppan = 763  &  irr = 90     : 881;
        P_out = 763  &   I_out = 204  &  Iout_conv = 204  &  I_L = 216 &  Ipan = 216  &  Vpan = 353 &  Ppan = 763  &  irr = 100     : 880;
        P_out = 826  &   I_out = 209  &  Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &  Vpan = 343 &  Ppan = 825  &  irr = 0     : 0;
        P_out = 826  &   I_out = 209  &  Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &  Vpan = 343 &  Ppan = 825  &  irr = 10     : 103;
        P_out = 826  &   I_out = 209  &  Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &  Vpan = 343 &  Ppan = 825  &  irr = 20     : 501;
        P_out = 826  &   I_out = 209  &  Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &  Vpan = 343 &  Ppan = 825  &  irr = 30     : 763;
        P_out = 826  &   I_out = 209  &  Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &  Vpan = 343 &  Ppan = 825  &  irr = 40     : 826;
        P_out = 826  &   I_out = 209  &  Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &  Vpan = 343 &  Ppan = 825  &  irr = 50     : 855;
        P_out = 826  &   I_out = 209  &  Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &  Vpan = 343 &  Ppan = 825  &  irr = 60     : 869;
        P_out = 826  &   I_out = 209  &  Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &  Vpan = 343 &  Ppan = 825  &  irr = 70     : 877;
        P_out = 826  &   I_out = 209  &  Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &  Vpan = 343 &  Ppan = 825  &  irr = 80     : 881;
        P_out = 826  &   I_out = 209  &  Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &  Vpan = 343 &  Ppan = 825  &  irr = 90     : 881;
        P_out = 826  &   I_out = 209  &  Iout_conv = 209  &  I_L = 241 &  Ipan = 241  &  Vpan = 343 &  Ppan = 825  &  irr = 100     : 880;
        P_out = 855  &   I_out = 213  &  Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &  Vpan = 348 &  Ppan = 853  &  irr = 0     : 0;
        P_out = 855  &   I_out = 213  &  Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &  Vpan = 348 &  Ppan = 853  &  irr = 10     : 103;
        P_out = 855  &   I_out = 213  &  Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &  Vpan = 348 &  Ppan = 853  &  irr = 20     : 497;
        P_out = 855  &   I_out = 213  &  Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &  Vpan = 348 &  Ppan = 853  &  irr = 30     : 763;
        P_out = 855  &   I_out = 213  &  Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &  Vpan = 348 &  Ppan = 853  &  irr = 40     : 825;
        P_out = 855  &   I_out = 213  &  Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &  Vpan = 348 &  Ppan = 853  &  irr = 50     : 855;
        P_out = 855  &   I_out = 213  &  Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &  Vpan = 348 &  Ppan = 853  &  irr = 60     : 869;
        P_out = 855  &   I_out = 213  &  Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &  Vpan = 348 &  Ppan = 853  &  irr = 70     : 877;
        P_out = 855  &   I_out = 213  &  Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &  Vpan = 348 &  Ppan = 853  &  irr = 80     : 881;
        P_out = 855  &   I_out = 213  &  Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &  Vpan = 348 &  Ppan = 853  &  irr = 90     : 882;
        P_out = 855  &   I_out = 213  &  Iout_conv = 213  &  I_L = 246 &  Ipan = 246  &  Vpan = 348 &  Ppan = 853  &  irr = 100     : 880;
        P_out = 869  &   I_out = 211  &  Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &  Vpan = 341 &  Ppan = 869  &  irr = 0     : 0;
        P_out = 869  &   I_out = 211  &  Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &  Vpan = 341 &  Ppan = 869  &  irr = 10     : 103;
        P_out = 869  &   I_out = 211  &  Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &  Vpan = 341 &  Ppan = 869  &  irr = 20     : 498;
        P_out = 869  &   I_out = 211  &  Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &  Vpan = 341 &  Ppan = 869  &  irr = 30     : 763;
        P_out = 869  &   I_out = 211  &  Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &  Vpan = 341 &  Ppan = 869  &  irr = 40     : 825;
        P_out = 869  &   I_out = 211  &  Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &  Vpan = 341 &  Ppan = 869  &  irr = 50     : 855;
        P_out = 869  &   I_out = 211  &  Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &  Vpan = 341 &  Ppan = 869  &  irr = 60     : 869;
        P_out = 869  &   I_out = 211  &  Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &  Vpan = 341 &  Ppan = 869  &  irr = 70     : 877;
        P_out = 869  &   I_out = 211  &  Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &  Vpan = 341 &  Ppan = 869  &  irr = 80     : 882;
        P_out = 869  &   I_out = 211  &  Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &  Vpan = 341 &  Ppan = 869  &  irr = 90     : 882;
        P_out = 869  &   I_out = 211  &  Iout_conv = 211  &  I_L = 254 &  Ipan = 254  &  Vpan = 341 &  Ppan = 869  &  irr = 100     : 880;
        P_out = 877  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Vpan = 342 &  Ppan = 877  &  irr = 0     : 0;
        P_out = 877  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Vpan = 342 &  Ppan = 877  &  irr = 10     : 104;
        P_out = 877  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Vpan = 342 &  Ppan = 877  &  irr = 20     : 497;
        P_out = 877  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Vpan = 342 &  Ppan = 877  &  irr = 30     : 763;
        P_out = 877  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Vpan = 342 &  Ppan = 877  &  irr = 40     : 826;
        P_out = 877  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Vpan = 342 &  Ppan = 877  &  irr = 50     : 855;
        P_out = 877  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Vpan = 342 &  Ppan = 877  &  irr = 60     : 869;
        P_out = 877  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Vpan = 342 &  Ppan = 877  &  irr = 70     : 877;
        P_out = 877  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Vpan = 342 &  Ppan = 877  &  irr = 80     : 881;
        P_out = 877  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Vpan = 342 &  Ppan = 877  &  irr = 90     : 881;
        P_out = 877  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Vpan = 342 &  Ppan = 877  &  irr = 100     : 880;
        P_out = 881  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  Ppan = 881  &  irr = 0     : 0;
        P_out = 881  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  Ppan = 881  &  irr = 10      : 104;
        P_out = 881  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  Ppan = 881  &  irr = 20      : 497;
        P_out = 881  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  Ppan = 881  &  irr = 30      : 763;
        P_out = 881  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  Ppan = 881  &  irr = 40      : 826;
        P_out = 881  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  Ppan = 881  &  irr = 50      : 855;
        P_out = 881  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  Ppan = 881  &  irr = 60      : 869;
        P_out = 881  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  Ppan = 881  &  irr = 70      : 877;
        P_out = 881  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  Ppan = 881  &  irr = 80      : 881;
        P_out = 881  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  Ppan = 881  &  irr = 90      : 881;
        P_out = 881  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  Ppan = 881  &  irr = 100      : 880;
        P_out = 881  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  Ppan = 882  &  irr = 0     : 0;
        P_out = 881  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  Ppan = 882  &  irr = 10      : 104;
        P_out = 881  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  Ppan = 882  &  irr = 20      : 497;
        P_out = 881  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  Ppan = 882  &  irr = 30      : 763;
        P_out = 881  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  Ppan = 882  &  irr = 40      : 826;
        P_out = 881  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  Ppan = 882  &  irr = 50      : 855;
        P_out = 881  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  Ppan = 882  &  irr = 60      : 869;
        P_out = 881  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  Ppan = 882  &  irr = 70      : 877;
        P_out = 881  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  Ppan = 882  &  irr = 80      : 881;
        P_out = 881  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  Ppan = 882  &  irr = 90      : 881;
        P_out = 881  &   I_out = 213  &  Iout_conv = 213  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  Ppan = 882  &  irr = 100      : 880;
        P_out = 880  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  Ppan = 880  &  irr = 0     : 0;
        P_out = 880  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  Ppan = 880  &  irr = 10      : 104;
        P_out = 880  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  Ppan = 880  &  irr = 20      : 497;
        P_out = 880  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  Ppan = 880  &  irr = 30      : 763;
        P_out = 880  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  Ppan = 880  &  irr = 40      : 826;
        P_out = 880  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  Ppan = 880  &  irr = 50      : 855;
        P_out = 880  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  Ppan = 880  &  irr = 60      : 869;
        P_out = 880  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  Ppan = 880  &  irr = 70      : 877;
        P_out = 880  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  Ppan = 880  &  irr = 80      : 881;
        P_out = 880  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  Ppan = 880  &  irr = 90      : 881;
        P_out = 880  &   I_out = 212  &  Iout_conv = 212  &  I_L = 257 &  Ipan = 257  &  Vpan = 343 &  Ppan = 880  &  irr = 100      : 880;
        TRUE                        : P_out;
    esac;
```

Bibliography

[1] R. Baheti and H. Gill, “Cyber-physical systems,” *The impact of control technology*, vol. 12, no. 1, pp. 161–166, 2011.

[2] J. Shi, J. Wan, H. Yan, and H. Suo, “A survey of cyber-physical systems,” in *Wireless Communications and Signal Processing (WCSP), 2011 International Conference on.* IEEE, 2011, pp. 1–6.

[3] S. Perry, “Model based design needs high level synthesis - a collection of high level synthesis techniques to improve productivity and quality of results for model based electronic design,” *2009 Design, Automation Test in Europe Conference Exhibition*, 2009.

[4] J. Nibert, M. Herniter, and Z. Chambers, “Model-based system design for mil, sil, and hil,” *World Electric Vehicle Journal*, vol. 5, no. 4, p. 1121–1130, 2012.

[5] M. S. Martis, “Validation of simulation based models: a theoretical outlook,” *The electronic journal of business research methods*, vol. 4, no. 1, pp. 39–46, 2006.

[6] V. D’silva, D. Kroening, and G. Weissenbacher, “A survey of automated techniques for formal software verification,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 7, pp. 1165–1178, 2008.

[7] P. Cousot and R. Cousot, “Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints,” in *Proceedings of the 4th ACM SIGACT-SIGPLAN symposium on Principles of programming languages.* ACM, 1977, pp. 238–252.

[8] R. Clarisó and J. Cortadella, "The octahedron abstract domain," in *International Static Analysis Symposium.* Springer, 2004, pp. 312–327.

[9] A. Deutsch, "Static verification of dynamic properties," in *ACM SIGAda 2003 Conference*, 2003.

[10] A. Miné, "The octagon abstract domain," *Higher-order and symbolic computation*, vol. 19, no. 1, pp. 31–100, 2006.

[11] B. Blanchet, P. Cousot, R. Cousot, J. Feret, L. Mauborgne, A. Miné, D. Monniaux, and X. Rival, "Design and implementation of a special-purpose static program analyzer for safety-critical real-time embedded software," in *The essence of computation.* Springer, 2002, pp. 85–108.

[12] M. T. Zuber, S. C. Solomon, R. J. Phillips, D. E. Smith, G. L. Tyler, O. Aharonson, G. Balmino, W. B. Banerdt, J. W. Head, C. L. Johnson *et al.*, "Internal structure and early thermal evolution of mars from mars global surveyor topography and gravity," *science*, vol. 287, no. 5459, pp. 1788–1793, 2000.

[13] "Static code analysis." [Online]. Available: https://www.mathworks.com/discovery/static-code-analysis.html

[14] "Synopsys software integrity." [Online]. Available: http://www.coverity.com/

[15] "Klocwork." [Online]. Available: http://www.klocwork.com/

[16] R. Huuck, A. Fehnker, S. Seefried, and J. Brauer, "Goanna: Syntactic software model checking," in *International Symposium on Automated Technology for Verification and Analysis.* Springer, 2008, pp. 216–221.

[17] K. R. M. Leino, "Efficient weakest preconditions," *Information Processing Letters*, vol. 93, no. 6, pp. 281–288, 2005.

[18] K. R. M. Leino and W. Schulte, "A verifying compiler for a multithreaded object-oriented," *Software System Reliability and Security*, vol. 9, p. 351, 2007.

[19] E. Ábrahám-Mumm, U. Hannemann, and M. Steffen, “Verification of hybrid systems: Formalization and proof rules in pvs,” in *Proceedings Seventh IEEE International Conference on Engineering of Complex Computer Systems.* IEEE, 2001, pp. 48–57.

[20] N. S. Bjørner, Z. Manna, H. B. Sipma, and T. E. Uribe, “Deductive verification of real-time systems using step,” in *International AMAST Workshop on Aspects of Real-Time Systems and Concurrent and Distributed Software.* Springer, 1997, pp. 22–43.

[21] T. Nipkow, L. C. Paulson, and M. Wenzel, *Isabelle/HOL: a proof assistant for higher-order logic.* Springer Science & Business Media, 2002, vol. 2283.

[22] G. Klein, K. Elphinstone, G. Heiser, J. Andronick, D. Cock, P. Derrin, D. Elkaduwe, K. Engelhardt, R. Kolanski, M. Norrish *et al.*, “sel4: Formal verification of an os kernel,” in *Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles.* ACM, 2009, pp. 207–220.

[23] G. Klein, J. Andronick, K. Elphinstone, T. Murray, T. Sewell, R. Kolanski, and G. Heiser, “Comprehensive formal verification of an os microkernel,” *ACM Transactions on Computer Systems (TOCS)*, vol. 32, no. 1, p. 2, 2014.

[24] L. Pike, S. Niller, and N. Wegmann, “Runtime verification for ultra-critical systems,” in *International Conference on Runtime Verification.* Springer, 2011, pp. 310–324.

[25] C. Daws, A. Olivero, S. Tripakis, and S. Yovine, “The tool kronos,” *International Hybrid Systems Workshop*, Oct 1995.

[26] K. G. Larsen, P. Pettersson, and W. Yi, “Uppaal in a nutshell,” *International Journal on Software Tools for Technology Transfer*, vol. 1, no. 1-2, p. 134–152, 1997.

[27] M. Ben-Ari and M. Ben-Ari, *Principles of the Spin Model Checker.* Springer London, 2008.

[28] T. A. Henzinger, P.-H. Ho, and H. Wong-Toi, “Hytech: a model checker for hybrid systems,” *International Journal on Software Tools for Technology Transfer*, vol. 1, no. 1-2, p. 110–122, 1997.

[29] A. Hinton, M. Kwiatkowska, G. Norman, and D. Parker, "Prism: A tool for automatic verification of probabilistic systems," *Tools and Algorithms for the Construction and Analysis of Systems Lecture Notes in Computer Science*, p. 441–444, 2006.

[30] H. Garavel, R. Mateescu, F. Lang, and W. Serwe, "Cadp 2006: A toolbox for the construction and analysis of distributed processes," *Computer Aided Verification Lecture Notes in Computer Science*, p. 158–163.

[31] R. Colgren, "Efficient model reduction for the control of large-scale systems," *Efficient Modeling and Control of Large-Scale Systems*, p. 59–72, 2010.

[32] Z. Han and B. Krogh, "Reachability analysis of hybrid control systems using reduced-order models," *Proceedings of the 2004 American Control Conference*, 2004.

[33] A. Tiwari and G. Khanna, "Series of abstractions for hybrid automata," *Hybrid Systems: Computation and Control Lecture Notes in Computer Science*, p. 465–478, 2002.

[34] T. Ball, V. Levin, and S. K. Rajamani, "A decade of software model checking with slam," *Communications of the ACM*, vol. 54, no. 7, p. 68, Jan 2011.

[35] E. M. Clarke and P. Zuliani, "Statistical model checking for cyber-physical systems," *Automated Technology for Verification and Analysis Lecture Notes in Computer Science*, p. 1–12, 2011.

[36] D. L. Parnas, "Really rethinking formal methods," *Computer*, vol. 43, no. 1, p. 28–34, 2010.

[37] T. Chen, M. Diciolla, M. Kwiatkowska, and A. Mereacre, "Quantitative verification of implantable cardiac pacemakers over hybrid heart models," *Information and Computation*, vol. 236, p. 87–101, 2014.

[38] G. Frehse, A. Hamann, S. Quinton, and M. Woehrle, "Formal analysis of timing effects on closed-loop properties of control software," *2014 IEEE Real-Time Systems Symposium*, 2014.

[39] A. Bauer, M. Leucker, and C. Schallhart, "Runtime verification for ltl and tltl," *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 20, no. 4, p. 14, 2011.

[40] A. Pnueli, "The temporal logic of programs," *18th Annual Symposium on Foundations of Computer Science (sfcs 1977)*, 1977.

[41] L. Grunske and P. Zhang, "Monitoring probabilistic properties," in *Proceedings of the the 7th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering.* ACM, 2009, pp. 183–192.

[42] A. Pnueli, A. Zaks, and L. Zuck, "Monitoring interfaces for faults," *Electronic Notes in Theoretical Computer Science*, vol. 144, no. 4, pp. 73–89, 2006.

[43] U. Sammapun, I. Lee, and O. Sokolsky, "Rt-mac: Runtime monitoring and checking of quantitative and probabilistic properties," in *11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA'05).* IEEE, 2005, pp. 147–153.

[44] H. Hansson and B. Jonsson, "A logic for reasoning about time and reliability," *Formal aspects of computing*, vol. 6, no. 5, pp. 512–535, 1994.

[45] C. Baier, J.-P. Katoen, and H. Hermanns, "Approximative symbolic model checking of continuous-time markov chains," in *International Conference on Concurrency Theory.* Springer, 1999, pp. 146–161.

[46] A. P. Sistla, M. Žefran, and Y. Feng, "Runtime monitoring of stochastic cyber-physical systems with hybrid state," in *International Conference on Runtime Verification.* Springer, 2011, pp. 276–293.

[47] N. Kosmatov, G. Petiot, and J. Signoles, "An optimized memory monitoring for runtime assertion checking of c programs," in *International Conference on Runtime Verification.* Springer, 2013, pp. 167–182.

[48] F. Chen and G. Roşu, "Java-mop: A monitoring oriented programming environment for java," in *International Conference on Tools and Algorithms for the Construction and Analysis of Systems.* Springer, 2005, pp. 546–550.

[49] M. V. Osch, "Automated model-based testing of simulation models with torx," *Lecture Notes in Computer Science Quality of Software Architectures and Software Quality*, p. 227–241, 2005.

[50] "The successful development process with matlab simulink in the framework of esas atv project," *55th International Astronautical Congress of the International Astronautical Federation, the International Academy of Astronautics, and the International Institute of Space Law*, Apr 2004.

[51] H. Elmqvist, M. Otter, D. Henriksson, B. Thiele, and S. E. Mattsson, "Modelica for embedded systems," *Proceedings of the 7 International Modelica Conference Como, Italy*, Aug 2009.

[52] G. Lipovszki and P. Aradi, "Simulating complex systems and processes in labview," *Journal of Mathematical Sciences*, vol. 132, no. 5, p. 629–636, 2006.

[53] A. Basu, M. Bozga, and J. Sifakis, "Modeling heterogeneous real-time components in bip," *Fourth IEEE International Conference on Software Engineering and Formal Methods (SEFM06).*

[54] J. Sifakis, "A framework for component-based construction," *Third IEEE International Conference on Software Engineering and Formal Methods (SEFM05)*, 2005.

[55] J. Botaschanjan, M. Broy, A. Gruler, A. Harhurin, S. Knapp, L. Kof, W. Paul, and M. Spichkova, "On the correctness of upper layers of automotive systems," *Formal Aspects of Computing*, vol. 20, no. 6, p. 637–662, 2008.

[56] N. He, P. Rümmer, and D. Kroening, "Test-case generation for embedded simulink via formal concept analysis," *Proceedings of the 48th Design Automation Conference - DAC 11*, 2011.

[57] Y. Annpureddy, C. Liu, G. Fainekos, and S. Sankaranarayanan, "S-taliro: A tool for temporal logic falsification for hybrid systems," *Tools and Algorithms for the Construction and Analysis of Systems Lecture Notes in Computer Science*, p. 254–257, 2011.

[58] "Simulink design verifier," *Reconstructing an Image from Projection Data - MATLAB Simulink Example.* [Online]. Available: https://www.mathworks.com/products/sldesignverifier.html

[59] M. Otter, N. Thuy, D. Bouskela, L. Buffoni, H. Elmqvist, P. Fritzson, A. Garro, A. Jardin, H. Olsson, M. Payelleville, and et al., "Formal requirements modeling for simulation-based verification," *Proceedings of the 11th International Modelica Conference, Versailles, France, September 21-23, 2015*, 2015.

[60] Y. Gurevich, "Evolving algebras: An attempt to discover semantics," *Current Trends in Theoretical Computer Science*, p. 266–292, 1993.

[61] M. Y. Vardi and Y. Wolper, "An automata-theoretic approach to program verification," *In: Symposium on Logic in Computer Science (LICS'86)*, p. 332–345, 1986.

[62] K. Kundert and A. Sangiovanni-Vincentelli, "Simulation of nonlinear circuits in the frequency domain," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 5, no. 4, p. 521–535, 1986.

[63] R. Trinchero, I. S. Stievano, and F. G. Canavero, "Steady-state analysis of switching power converters via augmented time-invariant equivalents," *IEEE Transactions on Power Electronics*, vol. 29, no. 11, p. 5657–5661, 2014.

[64] E. M. Clarke, O. Grumberg, and D. A. Peled, *Model checking.* The MIT Press, 2001.

[65] G. J. Holzmann, *The SPIN model checker: Primer and reference manual.* Addison-Wesley Reading, 2004, vol. 1003.

[66] A. Cimatti, E. Clarke, F. Giunchiglia, and M. Roveri, "Nusmv: a new symbolic model checker," *International Journal on Software*

Tools for Technology Transfer (STTT), vol. 2, no. 4, p. 410–425, Jan 2000.

[67] A. W. Roscoe, *The theory and practice of concurrency.* Prentice Hall, 2000.

[68] D. Jackson, *Software abstractions.* MIT press Cambridge, 2006, vol. 2.

[69] S. Leue and G. J. Holzmann, "v-promela: A visual, object-oriented language for SPIN," in *2nd International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC '99), May 2-5, 1999, Saint Malo, France*, 1999, pp. 14–23. [Online]. Available: https://doi.org/10.1109/ISORC.1999.776345

[70] K. McMillan, "Symbolic model checking . ph.d. thesis: , carnegie mellon university ()," 1993.

[71] E. M. Clarke and E. A. Emerson, "Design and synthesis of synchronization skeletons using branching time temporal logic," in *Workshop on Logic of Programs.* Springer, 1981, pp. 52–71.

[72] A. Salehi Fathabadi, M. J. Butler, S. Yang, L. Maeda, J. Bantock, B. M. Al-Hashimi, and G. V. Merrett, "A model-based framework for software portability and verification in embedded power management systems," *Journal of Systems Architecture*, vol. 82, 12 2017.

[73] G. Adinolfi, N. Femia, G. Petrone, G. Spagnuolo, and M. Vitelli, "Energy efficiency effective design of dc/dc converters for dmppt pv applications," in *Industrial Electronics, 2009. IECON'09. 35th Annual Conference of IEEE.* IEEE, 2009, pp. 4566–4570.

[74] Y. Driouich, M. Parente, and E. Tronci, "Modeling cyber-physical systems for automatic verification," in *Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD), 2017 14th International Conference on.* IEEE, 2017, pp. 1–4.

[75] T. Esram and P. L. Chapman, "Comparison of photovoltaic array maximum power point tracking techniques," *IEEE Transactions on energy conversion*, vol. 22, no. 2, pp. 439–449, 2007.

[76] M. D. Cristofaro, G. D. Capua, N. Femia, G. Petrone, G. Spagnuolo, and D. Toledo, "Models and methods for energy productivity analysis of pv systems," *2015 IEEE 13th International Conference on Industrial Informatics (INDIN)*, 2015.

[77] "Modelica and the modelica association." [Online]. Available: https://www.modelica.org/

[78] J. Åkesson, M. Gäfvert, and H. Tummescheit, "Jmodelica an open source platform for optimization of modelica models," *6th Vienna International Conference on Mathematical Modelling - Vienna, Austria*, 01 2009.

[79] "Jmodelica.org." [Online]. Available: https://jmodelica.org/

[80] T. Mancini, F. Mari, A. Massini, I. Melatti, F. Merli, and E. Tronci, "System level formal verification via model checking driven simulation," in *International Conference on Computer Aided Verification.* Springer, 2013, pp. 296–312.

[81] E. M. Clarke, O. Grumberg, M. Minea, and D. Peled, "State space reduction using partial order techniques," *International Journal on Software Tools for Technology Transfer*, vol. 2, no. 3, pp. 279–287, 1999.

[82] L. Pernebo and L. Silverman, "Model reduction via balanced state space representations," *IEEE Transactions on Automatic Control*, vol. 27, no. 2, pp. 382–387, 1982.

[83] A. Ferrante, M. Napoli, and M. Parente, "Model checking for graded ctl," *Fundamenta Informaticae*, vol. 96, no. 3, pp. 323–339, 2009.

[84] M. Faella, M. Napoli, and M. Parente, "Graded alternating-time temporal logic," *Fundam. Inform.*, vol. 105, no. 1-2, pp. 189–210, 2010.

[85] B. Aminof, A. Murano, and S. Rubin, "Ctl* with graded path modalities," *Inf. Comput.*, vol. 262, no. Part, pp. 1–21, 2018.

[86] A. Ferrante, M. Memoli, M. Napoli, M. Parente, and F. Sorrentino, “A nusmv extension for graded-ctl model checking,” in *Computer Aided Verification, 22nd International Conference, CAV 2010, Edinburgh, UK, July 15-19, 2010. Proceedings*, 2010, pp. 670–673.

[87] L. Sorrentino, S. Rubin, and A. Murano, “Graded ctl* over finite paths,” in *Proceedings of the 19th Italian Conference on Theoretical Computer Science, Urbino, Italy, September 18-20, 2018.*, 2018, pp. 152–161.