A formal approach to PLD design: from simulation to laboratory test

S. CARMELI, F. CASTELLI DEZZA, F. MAPELLI, A. MONTI
Dipartimento di Elettrotecnica
Politecnico di Milano
Piazza Leonardo da Vinci 32
20133 Milano - ITALY

Abstract: - Programmable Logic Devices (PLDs) gained an important role in the design of converter modulators. This paper describes an original approach able to synthesize PLD analysis with plant simulation. The methodology is confirmed through laboratory experience.

Key-Words: - High Level Petri Nets, PLD, power converters

1 Introduction

Implementation of control algorithms for power electronics application is an interesting challenge for PLD technology.
The most important features are:

- possibility to program the chip on board using a PC serial interface (in-system programmability),
- possibility to integrate many functions in the same chip thanks to an high level of integration;
- possibility to easily modify the chip programming.

From the economic viewpoint, PLD solution is particularly interesting for two reasons:

- the limited number of components necessary to support the "central intelligence", such EPROM and RAM as the "software" is directly on the chip;
- the extremely easy PCB design for the low number of wired connections.

The application of PLD for current control algorithms in power converters have been already discussed in a recent paper [1]. In particular, an original approach to algorithm design using a preliminary simulation step has been described and validated through experimental results. The importance of Petri Nets for system validation has been viceversa widely expressed in [2].
Full hardware or mixed hardware/software solution have been designed introducing a formal approach to the definition of the complete control model.

Aim of this paper is the description of an integrated solution for simulation where the Petri Net model of the PLD can interact with the plant model.
The Petri Net model is built composing simple Petri Nets, see [2], within the Cabernet tool.
The Cabernet model is then exported to Simulink using a new tool called Cab2mat. A complete simulation session is then possible introducing the plant model in the same Simulink schema.
The procedure has been tested in a real application and laboratory results are compared with simulation analysis.
The performance of a full bridge current controlled inverter are described and documented through simulation and laboratory data.

2 A Petri Net based approach

In [3] the importance of Petri Nets for the description of electrical systems where different subsystems interact in an unpredictable way is illustrated.

The most interesting features of Petri Nets and in particular of High Level Timed Petri Nets are [4]:

- possibility of describing concurrent systems (this is fundamental for mixed HW-SW solutions);
- possibility of completely validating the system verifying the "properties" of the net using automatic construction of the Network Reachability Tree;

- possibility of timing analysis when systems with different characteristics in terms of time-response interact.

The availability of computer aided tools for the analysis of Petri Nets (e. g. Cabernet) allows to easily perform all the analysis described in what above reaching an high level of reliability on the system. In what follows a Matlab/Cabernet integrated approach is presented through a real application example, so that also the plant model is included. The methodology is presented describing a real application.

3 A Real Application: a programmable current source.

A single phase inverter has been designed applying the original approach. The system is characterised by a digital current regulator managing also dead-time control (Figure 1).

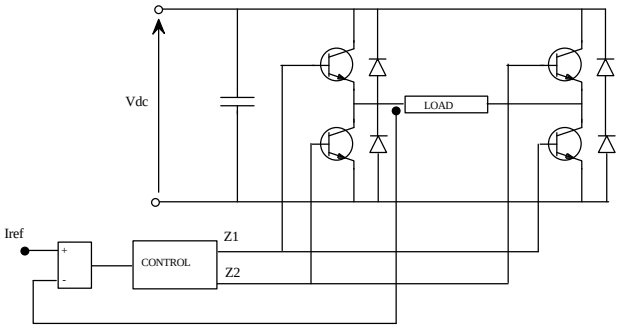

Figure 1: System architecture

Let us concentrate on the digital control system. The system clock is 1 Mhz. System output depends on the sign of the current error, i.e. the difference between the reference value and the actual current value. In particular two different situations can be evidenced:

1. The error is positive then choose +Vdc
2. The error is negative then choose -Vdc

The transition from one state to another is filtered for the dead time management. Let us suppose to acquire the reference from a waveform generator.
This analog signal is compared with the feedback load current. The difference is sampled using a 8-bit ADC operating at 1 MHz. The MSB can be applied as input for the control algorithm.
The control can be designed in terms of Finite State Machine when we have:

- one bit as input representing the signal of the control error
- two bits as output representing the desired state for the power switches.

In particular, Figure 2, illustrates the solution adopted.
Two states are introduced for normal conduction, and 4 for the transition and dead time management.
Let us suppose to be in the state (0,0,0), i.e. the one in the upper position.
If the error keeps the same sign (x = 0), the state remains the same. If the error turns to 1, the new state is (0,0,1) and all the switches are off.
If the error now keeps the sign the new state is (0,1,0) and then (1,0,0) where a new output platform for the switches is applied.
Applying this logic we generate a dead band between switch off and switch on equal to 3 μs.
The same considerations can be applied for the analysis for the left part of the FSM, moving from the state on the lowest position to the one in the upper position.

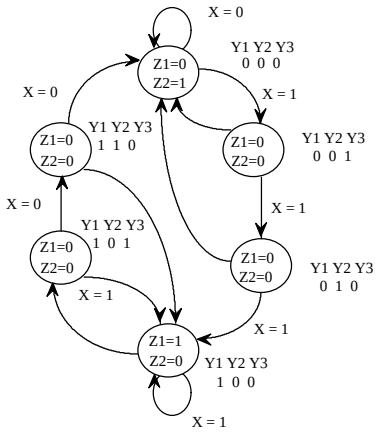

Figure 2: Finite state machine

The algorithm can be then translated into a logic net as the one reported in Figure 3. The second step consists in modeling the logic net through a High Level Petri Net.
Following the approach described in [3], every logic block can be mapped to a small HLTPN. The whole system is then obtained composing and synchronising the simple parts.
It should be noted that the definition of control algorithms requires a limited number of logic functions (AND, OR, NOT…) so that it is possible to develop a standard library and define any solution as a composition of these standard blocks.

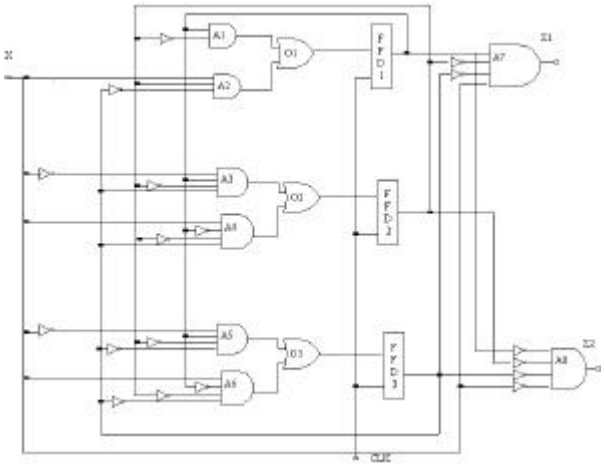

Figure 3: Logic net algorithm

The whole Petri Net has been divided into two separate nets with different functionality (Figure 4 and Figure 6):

1. the first Petri Net models signal conditioning phase
2. the second Petri Net models the control logic.

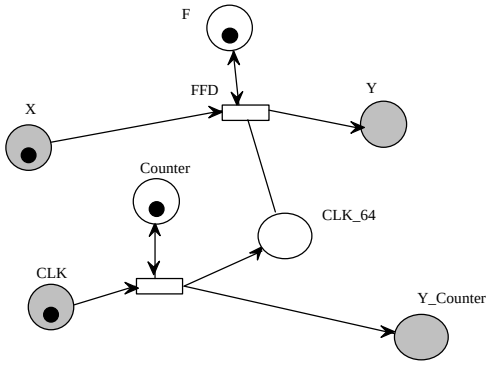

Figure 4: Signal conditioning Petri Net

For sake of clarity let us have a look to an example of block composing the control system.
Let us consider, e.g., the Petri Net model of a D Flip Flop (Figure 5); it is constituted of two input (IN and CLK that is the clock) and one output. Latch function is realised with a memory place (F).

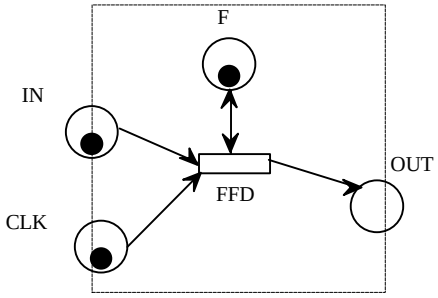

Figure 5: A flip flop Petri Net

The transition FFD masks the logic of the FF that will be described in terms of predicate and action.
In terms of pseudo code we have:

PREDICATE: TRUE (this means normal condition for enebling, i.e. all the input tokens should be present to fire the transition).

ACTION:
```
OUT.VALUE = F.VALUE
IF IN.VALUE = TRUE
    F.VALUE = TRUE
ELSE
    F.VALUE = FALSE
```

Each net can be easily modeled and verified within the Cabernet environment. The last step is the translation of the Cabernet file, that models the Petri net, into a Simulink block for time based simulation. This translation is possible thanks to a new and original tool (Cab2mat) that generates a Matlab executable file in DLL format.

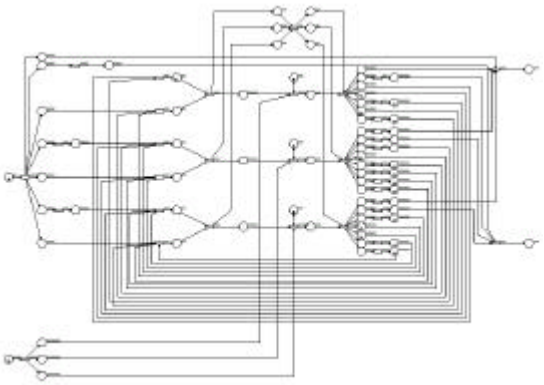

Figure 6: Digital control Petri Net

4 A new tool: Cab2mat

The key point of the proposed methodology is the automatic generation of a mixed model within the Simulink environment.
This opportunity is given by the new and original tool *Cab2mat*, through which the Petri Net model of the PLD is translated into a Simulink block.
In particular the Petri Net should be specified within the Cabernet tool. Only a restricted set of parameters is necessary for model exporting to Simulink:

- the initial values, i.e. the initial mark of the net
- the input places to link input places of the Petri Net to Simulink inputs;

- the output places to link output places of the Petri Net to Simulink outputs;

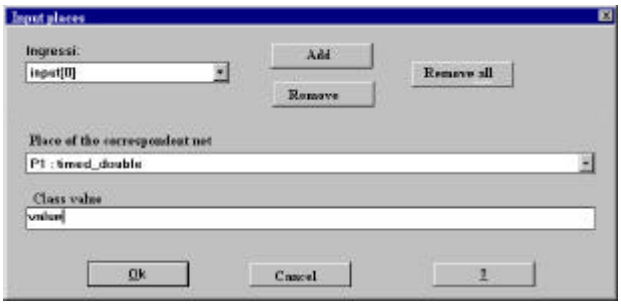

Figure 7: Input places assignment

A set of files is automatically generated from the Cabernet model (Figure 8):

1 a "DLL file", which is the executable Simulink file;
2 two ".m files", as interface;
3 a "cab2mat" file, to save the Petri Net data.

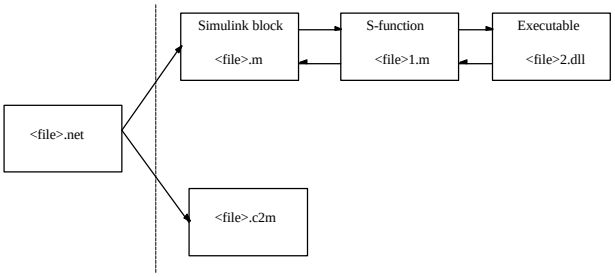

Figure 8: Files generated by Cab2mat

Once Simulink block is generated it can be easily integrated with the other parts of the system and a simulation phase can be started.

5 Experimental Results

The system here described has been realised and tested during a preliminary simulation phase and then using a laboratory prototype.
In particular the main data for the laboratory system are:

- 220 V, 50 Hz input voltage;
- 4 IGBT Semikron SKM 100 GB;
- Maximum output current: 20 A;
- Maximum output main frequency: 1 kHz;
- Custom board including PLD and auxiliary circuit for protection anf soft start.

During simulation phase the complete system has been tested in Simulink environment (Figure 9).

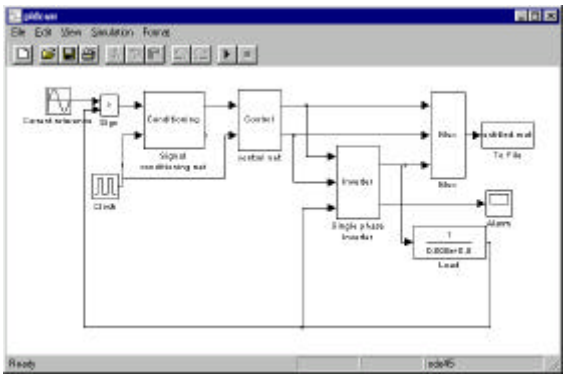

Figure 9: Simulink model of the system

The blocks "Conditioning" and "Control" are generated by Cab2mat tool and represent signal conditioning and digital control Petri Nets, respectively.
A comparison between the results obtained during the simulation phase and the ones obtained in laboratory confirms the efficiency of the design process in terms of low harmonic distortion. In particular an harmonic low impact on the load can be evidenced.
As it can be also deduced from the Simulink block, the load is a resistor of 0.8 Ω in series with an inductance of 8 mH.
Figure 10 and Figure 11 reports one simulation and laboratory test respectively, at 100 Hz with maximum value of current equal to 10 A.

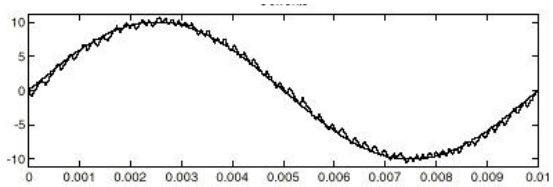

Figure 10: A simulation result with sinusoidal reference

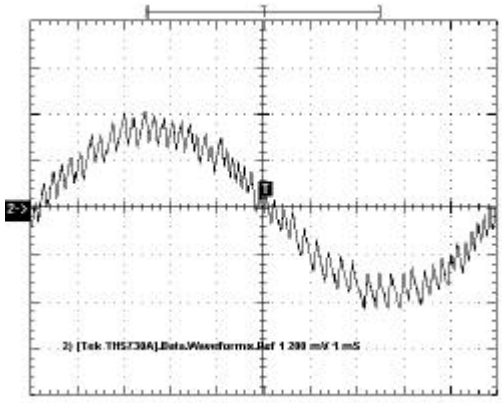

Figure 11: A laboratory result with sinusoidal reference

The performance of the dead-time logic for short circuit prevention in every inverter leg is analysed in Figure 12.A 2 μs dead-time has been realised according to the project requirements.

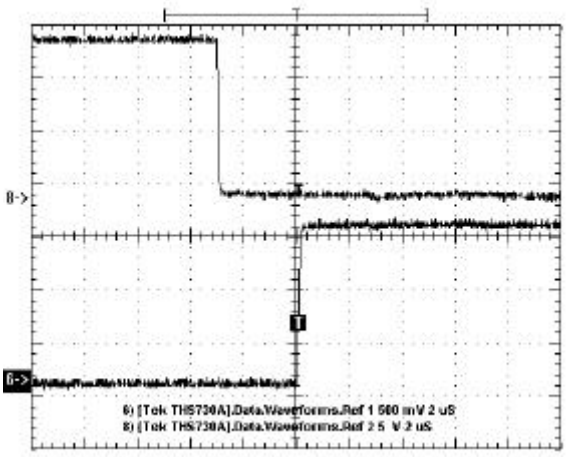

Figure 12: Dead time analysis

Figure 13 illustrates the possibility to synthesise waveforms different from the sinusoidal one. In this case we have a triangular current at 500 Hz. Other tests have been conducted at different frequencies applying different waveforms as input reference.

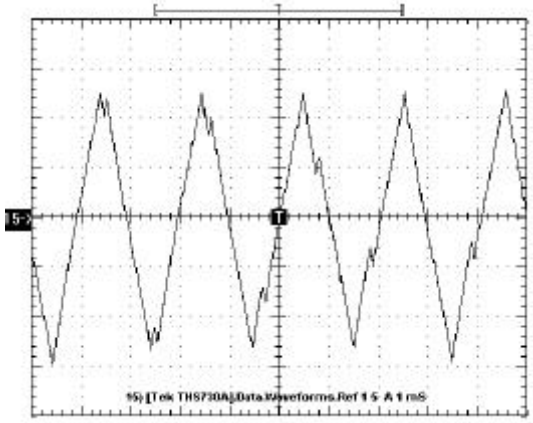

Figure 13: An experimental result with triangular reference

6 Conclusion

A new Petri Net based approach for PLD design has been presented. In particular the key point of the method is the translation of the Petri Net model in a Simulink block.

Thank to a new and original tool, Cab2mat, a Petri Net can be easily translated into a Simulink block and an integrated simulation is possible.

The approach has been tested and validated with good results either in simulation and with a laboratory prototype

Future extensions of this approach can be esily evidenced.

In effect Cab2mat is basically a C-code generator and then the model can be exported for other platforms simply changing the interface (this is represented by two external files and then do not influence the C code).

A petri Net based approach could be then applied also using different simulation environment(e.g. PSPICE).

In a few words, Cab2mat can be considered a model generator for PLD simulation every time we want to verify the interaction within complex systems.

7 References

[1] F. Castelli Dezza, M. Lazzaroni, A. Monti, Current Controlled Power Converters Through In System Programmable Devices, IEEE ICECS 97, Cairo (Egypt)

[2] A. Monti, Circuits as Concurrent Systems: a new approach to the simulation, IEEE ISCAS 96 Atalanta (USA)

[3] S. Carmeli, M. Lazzaroni, A. Monti, PLD Implementation of Control Algorithms: Design and Validation, IEEE ISCAS 98 USA

[4] T. Murata "Petri Nets: Properties, Analysis and applications", IEEE Proc., vol 77, n.4, pp 541-580, April 1989