PORTS

Creation Guide

Version #: **0.1**

Revision date: **30 / 11 / 2010**

Version Control

Revision	Authors	Date	Comments
0.1	Julierme Silva de Araujo Lizbeth L. Paredes A.	20/11/2010	Initial Version
0.2	Julierme Silva de Araujo Lizbeth L. Paredes A.	30/11/2010	Final version before of the revision (Complete version)

Table of Contents

PORTS 4

1.1 INTRODUCTION 4

1.2 MODULE HISTORY 4

1.3 FEATURES 4

1.4 DESIGN ASSUMPTIONS 5

1.5 ARCHITECTURE AND DESIGN HIERARCHY 6

1.6 PORTS DESCRIPTION 7

1.7 PORTS SIGNALS 7

1.8 PORTS THEORY OF OPERATION 8

1.9 PORTS IMPLEMENTATION 9

1.10 PORTS CONFIGURATION PARAMETERS 9

1.11 PORTS PERFORMANCE 11

1.12 PORTS DESIGN TRADE-OFFS 11

1.13 PORTS REUSE ISSUES 12

1.14 DESIGN METHODOLOGY 12

1.15 SIMULATION STRATEGY AND RESULTS 12

1.16 TOOLS FLOW 13

1.17 DESIGN FOR TEST 13

1.18 DESIGN FOR TEST 13

1.19 PATENT DISCLOSURES 13

PORTS

1.1 Introduction

The EMC08 microcontroller 8 bit is composed of several blocks like to see in the figure 1, one being the block ports that will be introduced in this document. The block ports consistency of the 4 ports of the 8 bits bidirectional (P0-P3) and one unidirectional port (P4) 8 bits too, like show in the figure 2. The port 3 (P3) is multifunctional and it can have several configurations.

All ports can be configured as inputs or outputs. Accordingly, in total of 32 input/output pins and more 8 outputs enabling the microcontroller to be connected to peripheral devices are available for use. Pin configuration, i.e. whether it is to be configured as an input (1) or an output (0), depends on its logic state defined for the P0EN-P3EN registers. In order to configure a pin microcontroller as an input, it is necessary to apply a zero logic (0) to appropriate I/O port bit. In this case, voltage level on appropriate pin will be 0.

1.2 Module History

Table 1 lists the history of the Ports module versions and implementations.

Table 1 – Module History Example

Module Version	Derived From	1st Implemented On	Headline
Ports(0.1)		Ports0.1(EMC08)	---
Ports(0.2)	Ports(0.1)	Ports0.2(EMC08)	Describe major changes from previous version
Ports(0.3)	Ports(0.2)	Ports0.3(EMC08)	Describe new changes from previous version
Ports(0.4)	Ports(0.3)	Ports0.4(EMC08)	Actualizations in the Ports Block access of the SFR was I/O, now there are one input for I and other input for O
Ports(0.5)	Ports(0.4)	Ports0.5(EMC08)	Actualizations in the Ports Block design - Mode Test

1.3 Features

Ports Block is connected right the blocks: interrupts, serial and Bus Control (access to external memory) and the SFR (Special Functions Registers). Each of these blocks will be responsible for setting the bits of configuration of each port like input or output and making possible the flow of data.

It can be seen in Table 2 records the settings of P [3:0] EN for each port to behave as I/O (input or output).

- Ports Block is a module completely passive (Combinational), them it is dependent on the changes performed by other modules system.
- It has the functionality to be prepared to configure the ports so that events involving the flow of data (Tx/Rx) for these ports will assist.
- The block ports are not functionality making any kind of access control and/or data type in for the itself.

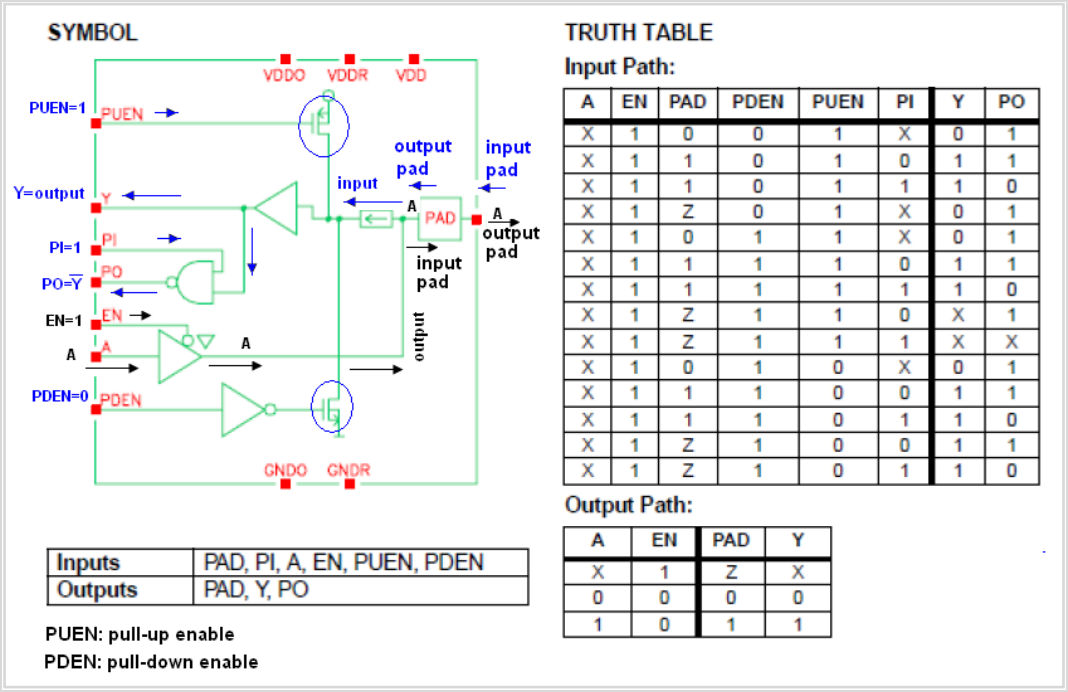
1.4 Design Assumptions

Assumptions for the creation

Consider that there is IP modules provided by the foundry itself to set the pins (PADs) of EMC08 chip, which were proposed by the project SOW.

Them considered the module of X-FAB library which are: (ddk/X-FAB-doc/XCOL8_IO_LIB_MAN_V3_1.pdf) model proposed in the SOW, which limits the operation of this block to configure the IP module.

Consider also that this module as shown in Figure 1. It has 12 inputs and 3 outputs where its configuration depends on whether the port will behave as input or output.



A	EN	PAD	PDEN	PUEN	PI	Y	PO
X	1	0	0	1	X	0	1
X	1	1	0	1	0	1	1
X	1	1	0	1	1	1	0
X	1	Z	0	1	X	0	1
X	1	0	1	1	X	0	1
X	1	1	1	1	0	1	1
X	1	1	1	1	1	1	0
X	1	Z	1	1	0	X	1
X	1	Z	1	1	1	X	X
X	1	0	1	0	X	0	1
X	1	1	1	0	0	1	1
X	1	1	1	0	1	1	0
X	1	Z	1	0	0	1	1
X	1	Z	1	0	1	1	0

A	EN	PAD	Y
X	1	Z	X
0	0	0	0
1	0	1	1

Figure 1 – I/O Circuits of the PADs

Is important know, that because of having of the threads of communication with the SFR, the only entry that was treated is the EN (Enable) what it defines, if EN=0 the port is shaped like exit like if is shown in the previous chart, if EN=1 the port is shaped like entry. All other values of configuration of the PADs will be defined in the Back-end since they are fixed values and will be tied to these values for security.

Typical application

The ports block is a simple module of configuration, which it defines if some bit of the registers P0EN to the P3EN is 0 , then EN (enable) of this bit to 0 and which configure the port like exit, while if some bit of the P0EN to the P3EN is 1, then EN (enable) of this bit to 1 and the port is configure like entry.

Typically the application would be in modules of control, of alone configuration.

Limitations on using the VC

In the development of the block ports it must be to think that the module is a controlador for the step of data of a module to any other, this module of only configuration is separated of the PADs and of the modules of where the data and the modules for where go the data. Then the design was limited to a similar characteristics, or it would have to be changed if the this module of configuration inside the same PADs in the case.

There was defined the module of configuration separated by characteristics of Back-end (roteamento) and easiness of implementation.

1.5 Architecture and Design Hierarchy

Como foi falado anteriormente, o block ports esta separado dos PADs configurados, então a arquitetura fica definida como se mostra na figura 2.

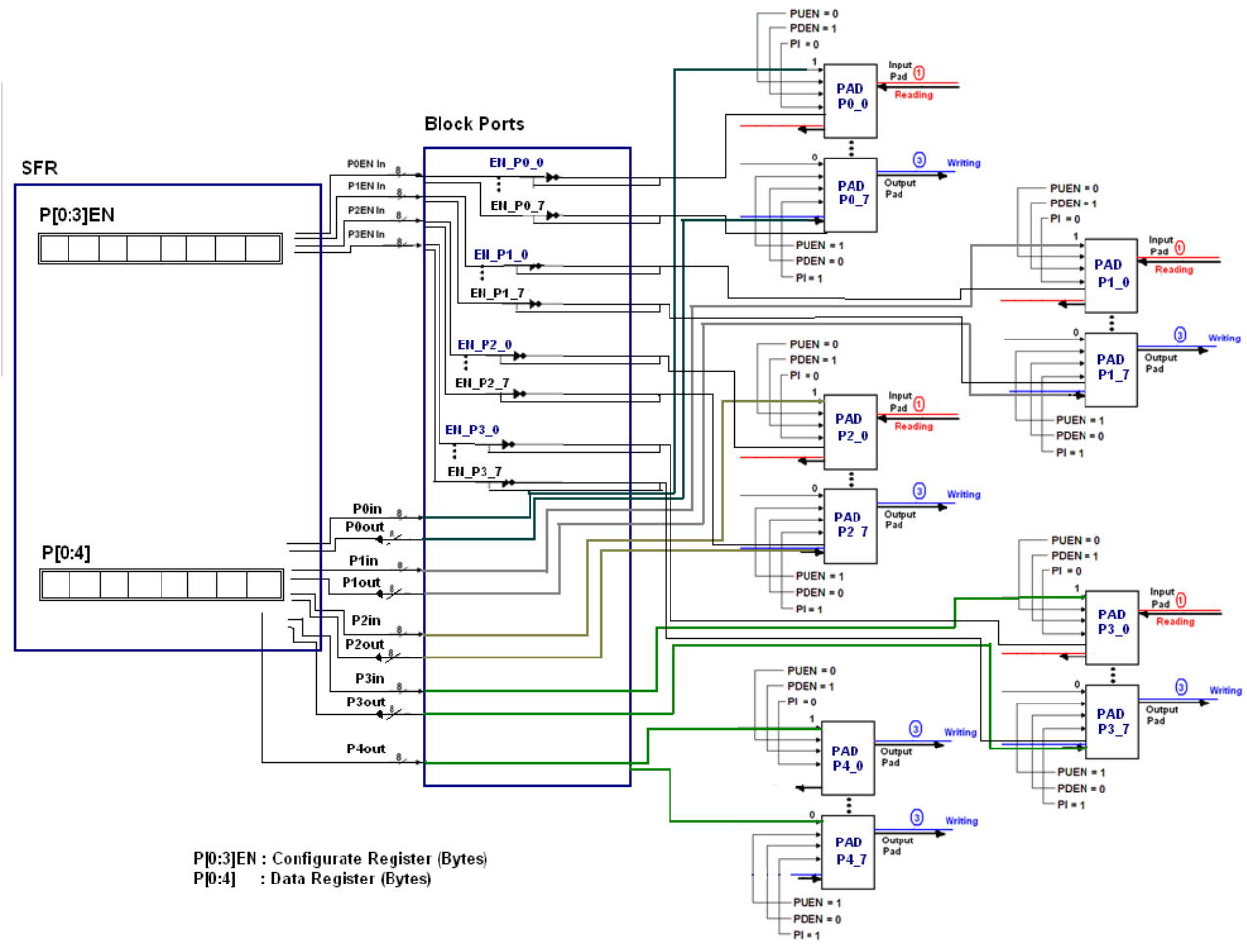


Figure 2 – Example for the Input/Output Configurations Port0

1.6 Ports Description

Setting: The block has the functionality to configure ports each pin input/output (I/O) to be used by different blocks in the Tx/Rx data information or addresses.
Control: Ports Block don't has control functionality, it is solely responsible for configuring the ports as I/O, their decision inputs or outputs will be the responsibility of other blocks (interrupts, Tx, Rx serial access memory).

The figure 3 is showed the several ports with their several functionalities.

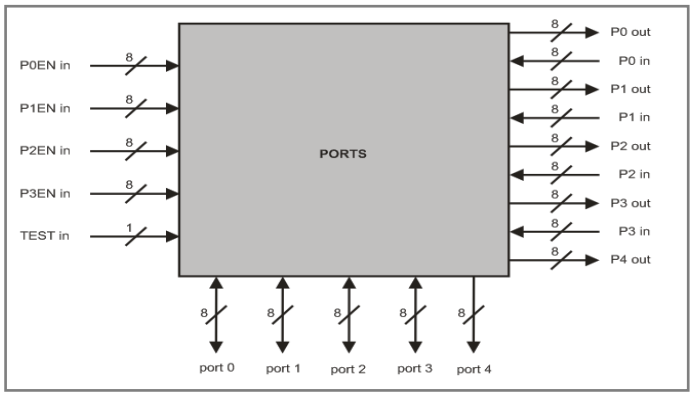


Figure 3 – Block Diagram

1.7 Ports Signals

Table 2 – Detailed signal descriptions

Registers	Pins	Bit (s)	I/O	Description
P0	[7:0]	8	I/O	Special Function Register for the store of data
P1	[7:0]	8	I/O	Special Function Register for the store of data
P2	[7:0]	8	I/O	Special Function Register for the store of data
P3	[7:0]	8	I/O	Special Function Register for the store of data
P4	[7:0]	8	I/O	Special Function Register for the store of data
P0EN	[7:0]	8	I	Enable port 0 (POEN=0 input RD) (POEN=1 output WR)
P1EN	[7:0]	8	I	Enable port 1 (P1EN=0 input RD) (P1EN=1 output WR)

P2EN	[7:0]	8	I	Enable port 2 (P2EN=0 input RD) (P2EN=1 output WR)
P3EN	[7:0]	8	I	Enable port 3 (P3EN=0 input RD) (P3EN=1 output WR)
Port O	[7:0]	8	I/O	RD/WR dados da memória externa
Port 1	[7:0]	8	I/O	RD/WR of generally proposed
Port 2	[7:0]	8	I/O	Higher address [15:8] external memory
Port 4	[7:0]	8	I/O	Lower address [7:0] bus output
Port 3	[7:0]	8	I/O	Generally proposed
	P3.0	1	I	RXD (serial input port)
	P3.1	1	I	TXD (serial output port)
	P3.2	1	I	INT1 (External interrup.)
	P3.3	1	I	INT0 (External interrup.)
	P3.4	1	I	Generally proposed
	P3.5	1	I	Generally proposed
	P3.6	1	I	WR_b external data memory write strobe
	P3.7	1	I	RD_b external data memory read strobe

1.8 Ports Theory of Operation

- When needed, access to the external memory, input buffers on port 0 and output drives to ports 0, 2, 4 are used:
  - In this case, the data bus in port 0 is bidirectional;
  - The port 2 has an output byte in the most significative address byte of the external memory when the address is 16 bits. In other words, the port 2 pins continue emitting the contents of SFR.
- The drive from the port 0 can be:
  - The bus data from external memory;
  - Or the I/O general purpose.
- The port 2 can be changed to:
  - The high address bus;
  - Or the I/O general purpose.
- The output drivers of Ports 0 can be switchable to DATA BUS or General Purpose I/O and Port 2 can be switchable to HIGHER ADDRESS BUS to General Purpose I/O by an internal CONTROL signal for its external memory access.
  - The port 4 is the address bus down. During the access to external memory the rest of special registers P0/P2 remain unchanged.
- Each port of the I/O can be independently used as input to the output by P [3:0] configured for records.

(Ports 0 and Ports 2 can't be used for I/O general purpose when it is used as address or data bus).

1.9 Ports Implementation

This module is completely combinational, it is very simple and it is described in the next iten.

1.10 Ports Configuration Parameters

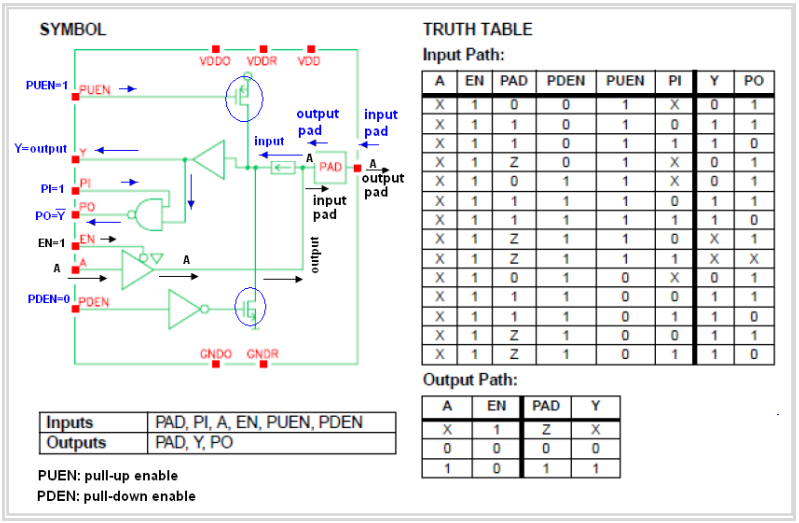
1.10.1 I/O PADs Configurations

The block ports have two configurations modes that can be input or output. These configurations are the following:

1.10.1.1 Considering that each port is as follows:

Figure 4 – Parts ports block

In the figure 5 are showed the diagram functional of a I/O buffer pins on each of the ports.



A	EN	PAD	PDEN	PUEN	PI	Y	PO
X	1	0	0	1	X	0	1
X	1	1	0	1	0	1	1
X	1	1	0	1	1	1	0
X	1	Z	0	1	X	0	1
X	1	0	1	1	X	0	1
X	1	1	1	1	0	1	1
X	1	1	1	1	1	1	0
X	1	Z	1	1	0	X	1
X	1	Z	1	1	1	X	X
X	1	0	1	0	X	0	1
X	1	1	1	0	0	1	1
X	1	1	1	0	1	1	0
X	1	Z	1	0	0	1	1
X	1	Z	1	0	1	1	0

A	EN	PAD	Y
X	1	Z	X
0	0	0	0
1	0	1	1

Figure 5 – I/O Circuits of the PADs

1.10.2 Circuit Configuration

1.10.2.1 Input/Output (I/O) pin:

How does work the input port microcontroller?

- When PUEN (Pull-up is in enable = 1) then the transistor is biased and allows the power circuit, and enables the output Y. In another circuit would be in open circuit.

- When PDEN (Pull-down is not in enable = 0) for the presence of the denied buffer transistor is polarized and closes the circuit.
  - The output Y can be used as denied that activating the input (PI = 1).
  - If PI = 0, only if Y is the output only.

1.10.2.2 Input pin:

A logic one (1) is applied to a bit of the P register. The output FE transistor is turned off and the appropriate pin remains connected to the power supply voltage over a pull-up resistor of high resistance.

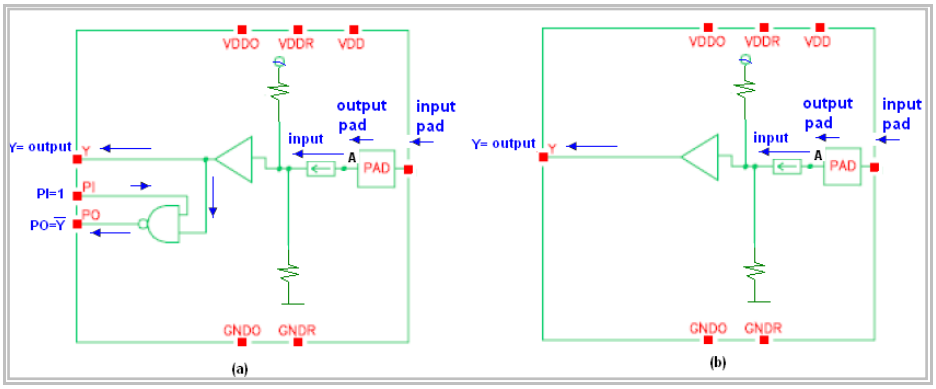


Figure 6 – Circuit for the Input PADs configurations

Table 3 – Inputs and Outputs for the INPUT PADs

Inputs	EN = 0 PUEN = 1 PDEN = 0 If PI = 1 them figure (a) If PI = 0 them figure (b)	**Output**	If PI = 1 them output = Y and PO If PI = 0 them output = Y

- **How does work the output port microcontroller?**

  - If EN = 0 in this case the output would be high impedance (Z).
  - If EN = 1 in the case allows the pitch of the A signal that is sent to the PAD.

Logic zero (0) is applied to a bit of the P register. The output EN transistor is turned on, thus connecting the appropriate pin to ground.

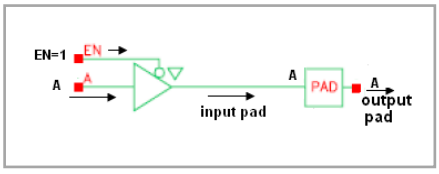


Figure 7– Circuit for the output configurations

Table 4 – Inputs and Ou1=tputs for the OUTPUT PADs

Inputs	EN = 1 PUEN = 0 PDEN = 1 A(Data)	**Outputs**	A (data)

The Ports Block will configure the PAD cells of each pin through pin EN of this cell. When the Ports Block configures the pin EN with value equal to 0 (zero) the PAD cell will be configured to receive data coming from the Ports Block through pin A. When the Ports Block configures the pin EN with value equal to 1 (one) the PAD cell will be configured to receive data from the external environment and send these data to the Ports Block through pin Y.

1.11 Ports Performance

The block Ports is a very simple block show 1.6 of this document was described in the topic and detailed in the topics that follow it. With the final version of the block it approached to a dry block and with high performance principally when necessary the parallelism of data through the ports.

1.12 Ports Design Trade-offs

The development of the block Ports took 2 modules that this are sub-modules of the module TOP. These sub-modules are the module Port that defines the doors P0, P1, P2 and P3 and the sub-module Port_4 that defines the door P4. It was necessary a module to part for the door P4 because it carries this one has a very specific and limited functionality.

Four ports of the EMC08 are bidirectional (P0-P3). Each of these ports consists of a latch (Special Function Registers P0 through P3), an output driver and an input buffer. Port 4 is the output of the address bus. The I/O ports (P0-P3) are bit configured by SRF registers (P[3:0]EN) and the P3.1 has an special bit configuration in PCON register (P3SEL) to select the output source from TXD (Serial) or P3.1 register.

Hence, each port can be configured as input or output, them this module can be considered as a sub-module of the ports block like show in the figure 9.

The figure 9 shows an example of the ports module configuration, with pad 0 as input and pad 7 as output for port 0. It indicates the right values for EN, PUEN, PDEN, PI, etc. As well as, the figure 7 and 8 and tables 5 and 6 show the same information.

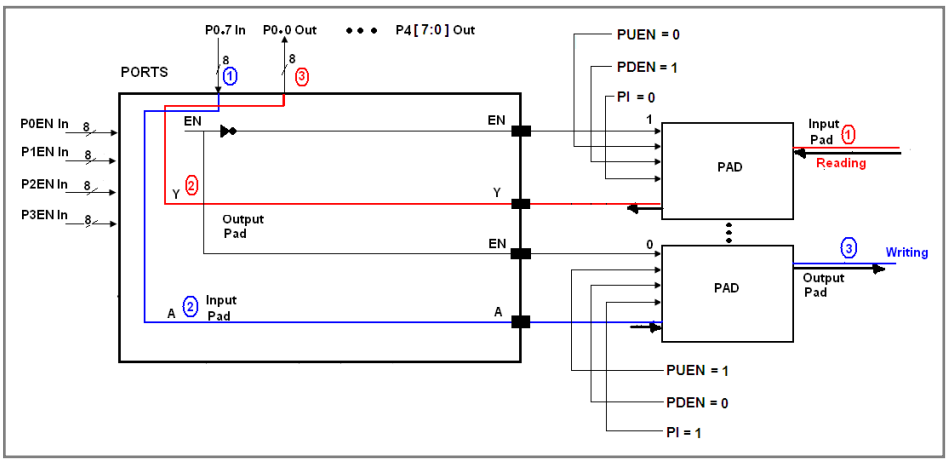


Figure 9 – Example for the Input/Output Configurations Port0

1.13 Ports Reuse Issues

There were some problems in ancient versions of the block Ports due to the adaptation of this block the changes that took place in the architecture of the EMC08. While the architecture of the chip was modified the alterations in the block Ports they were necessary aiming adapting this block the evolutions implemented by the team of the project EMC08.

1.14 Design Methodology

The design methodology used in this project was the functionality development in HDL tools in this case Verilog .

Because this module to be only configuration for the ports, this module is only combinational, It don't depended for the clock then this module don't have problems for the time.

1.15 Simulation Strategy and Results

For the tests and analyses was development testsbench file, for the three important test:

- Configuration ports test
- Input random test for the 40 ports (pins).
- Output random test for the 40 ports (pins).

All results were positive like the show in the figure 10.

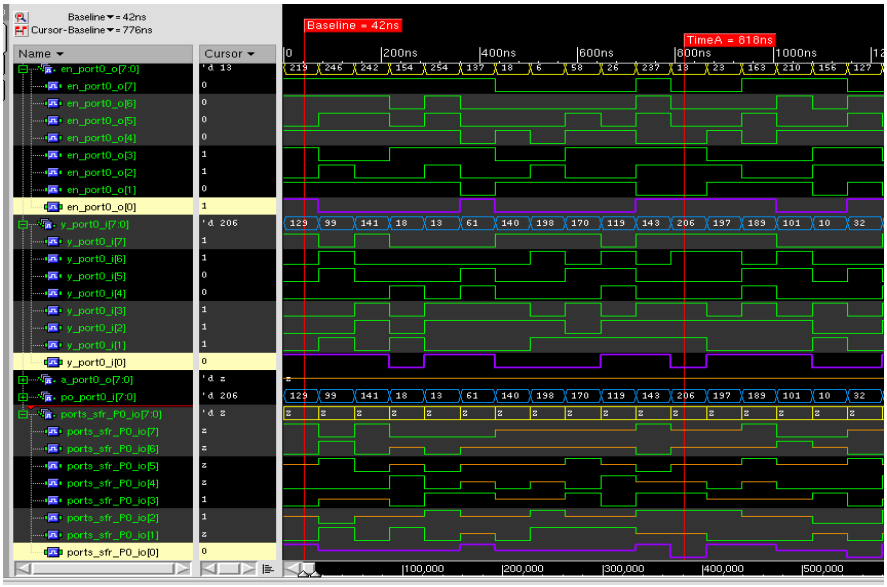


Figure 10 – Example for the Read Port0

If *en_port0_o[0]=1* like in this case, then the data *y_port0_i[0] = 0* , it puts this data 0 in the SFR *ports_sfr_P0_io[0].*

1.16 Tools Flow

- All the modules were developed in VERILOG with the CADENCE tools.
- The design was tested with Testbench.
- Summarized with the tool Encounter (for to value LATCHs, turn back prior of Area, Power, Timing, sequential report), when the Netlist (Of the synthesis) was produced.
- LEC Analysis (for to check that the whole code VERILOG is summarized (Comparison of the GOLDEN (RTL) and the REVISED (NETLIST)). For these is need the Encounter Tools .

- for the verification, all the modules must be checked, for this each person in charge has to prepare the Verification guide, where there are indicated the features that to be valued of the checking. For these was development in e language (Specman).

1.17 Design for Test

For Desig for test in the block ports is module combinational, this module was tested for the DFT of the TOP.

1.18 Design for Test

For Desig for test in the block ports is module combinational, this module was tested for the DFT of the TOP.

1.19 Patent Disclosures

In this case it is obvious makes mark what is not good a bidirectional thread uses is better two threads have for cases of pins or carry it bidirectional.