Politecnico di Torino

Switching Technologies for Data Centers

Laboratory

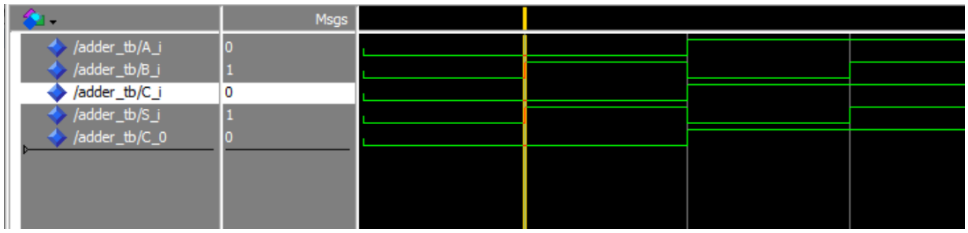
2021

Professor Marco VACCA

Wojciech Paluszkiewicz, s293958

Lab.1

1.1 Implementation of a Full-Adder



According to the rule:

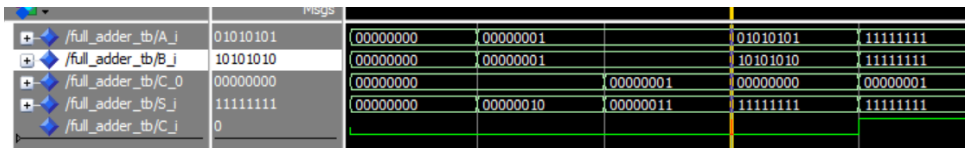
$$S = (A \oplus B) \oplus C_i$$

$$C_o = (A \cdot B) + (C_i \cdot (A \oplus B))$$

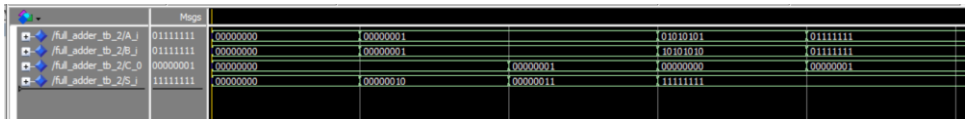
S is a sum and C0 is carry value from this operation.

Ci is a input carry vaule from the operation before.

1.2 Implementation of a Ripple Carry Adder and comparison with "+" operator



Second test bench:



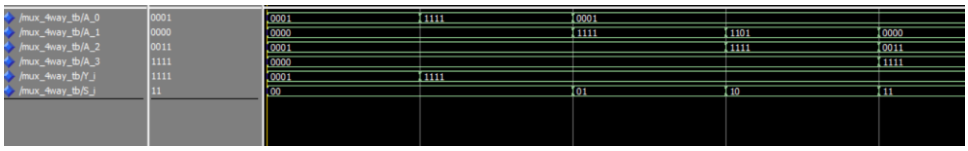
A, B – values

C_0 - carry bit defined by user to check behaviour

S_i – result of summing

In this case, I am adding two 8-bits value – A and B. C0 is a vector representing a carried bit from the right (outside). Ci is a carry value from this summing.

1.3 Implementation of a 4–way multiplexer



A_0, A_1, A_2, A_3 – values for inputs of multiplexer

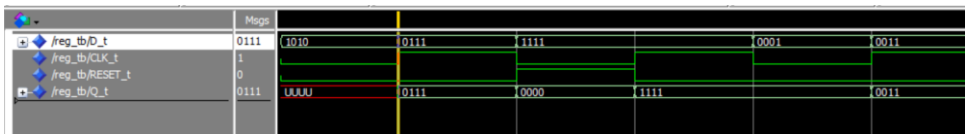
Y_i – output, the same size as each input

S_i – selector for MUX

In this case generic size of input and output is 4.
S_i is my selector in the multiplexer.

As an output we have input selected by S_i input.

1.4 Implementation of a register



D_t – input vector

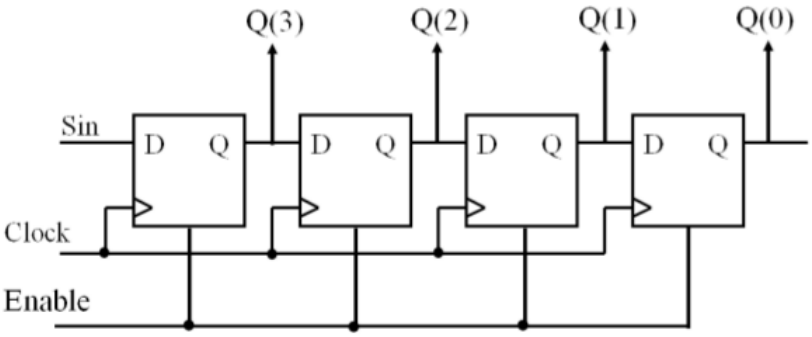
RESET_t – asynchronous reset active high

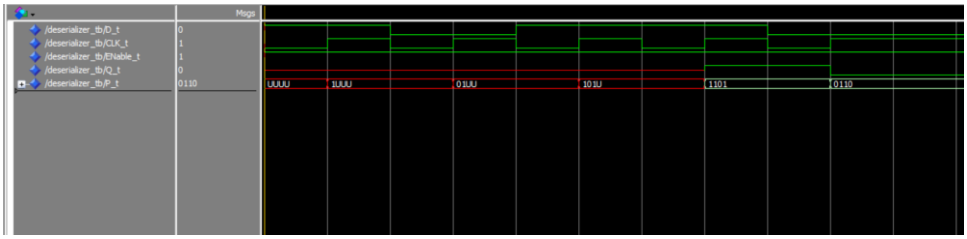
Q_t – register's output

The register is a positive-edge triggered, so it means, that every rising edge of the clock an input D_t is copied to output Q_t. This state is stable during the duration of one clock cycle (from a rising edge to the next positive clock's edge). It has a generic size N =4.

1.5 Implementation of a serializer and a deserializer

1.5.1





D_t – one bit input

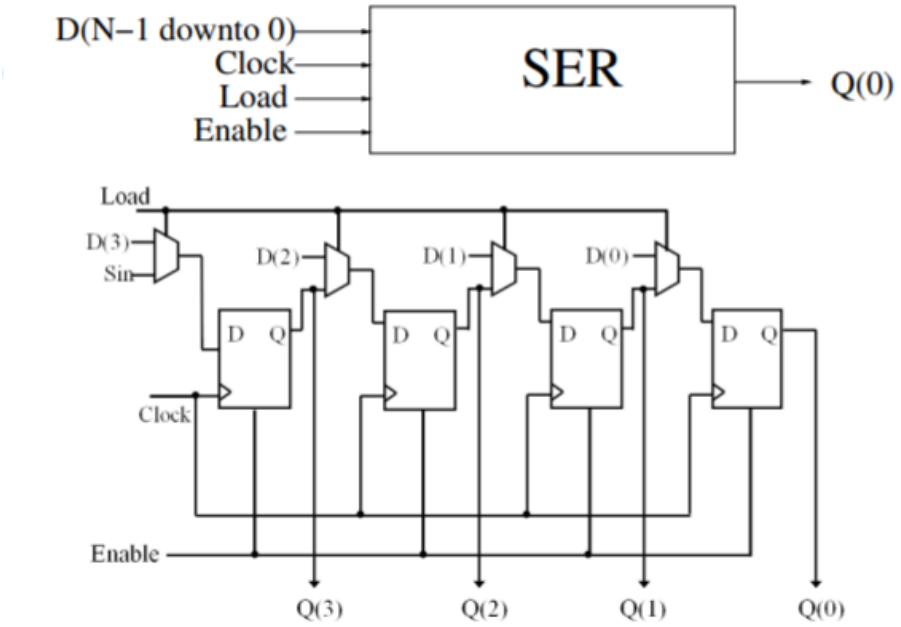
Enable_t – allow to run circuit

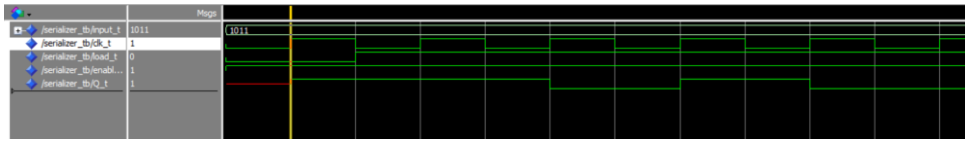
Q_t – output, in which the oldest value (one bit) is placed

P_t – internal vector to store given bits

My deserializer takes input-D_t and in the case of the clock's positive edge, it moves this value to the output vector on the first position from left. Then output vector waits for the next value, when it occurs(new value and positive edge), the vector's values shift one position to the right.

1.5.2





Input_t – vector for input

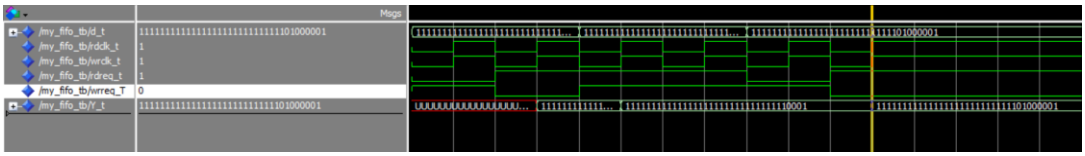
Load_t – when is ‘0’ the input vector loads to registers

Q_t – each rising edge of clock new output bit appears

Serializer, when the load is “0” and a positive edge of a clock, the registers are filled with the input vector. Subsequently, output Q_t takes bit after bit from the right side from the registers filled by input on every positive clock’s edge.

Lab.2

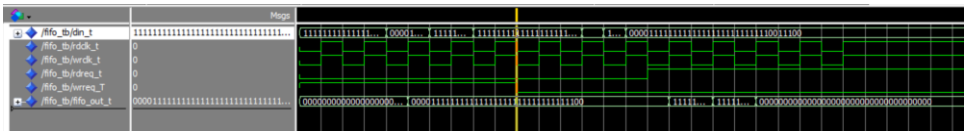
My fifo:



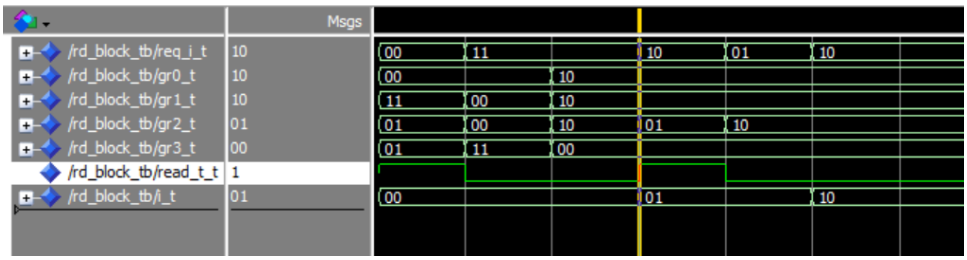
When wrreq_T is 1 and wrclk's positive edge events, the circuit saves d_t vectors in the internal registers.

If rdreq is 1 and rdclk's has a positive edge, the circuit assigns to output internal register's values.

Given FIFO:

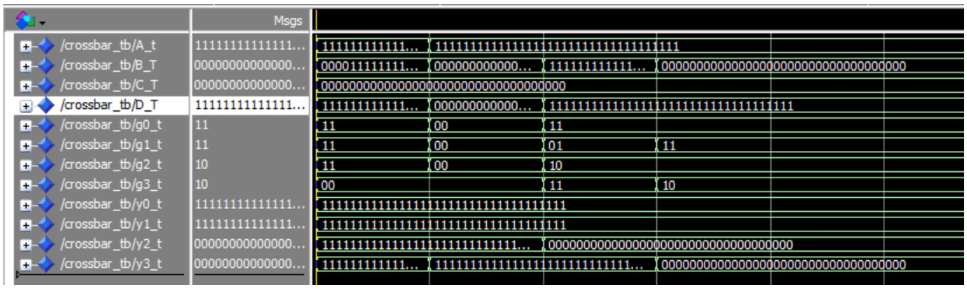


RD_BLOCK:

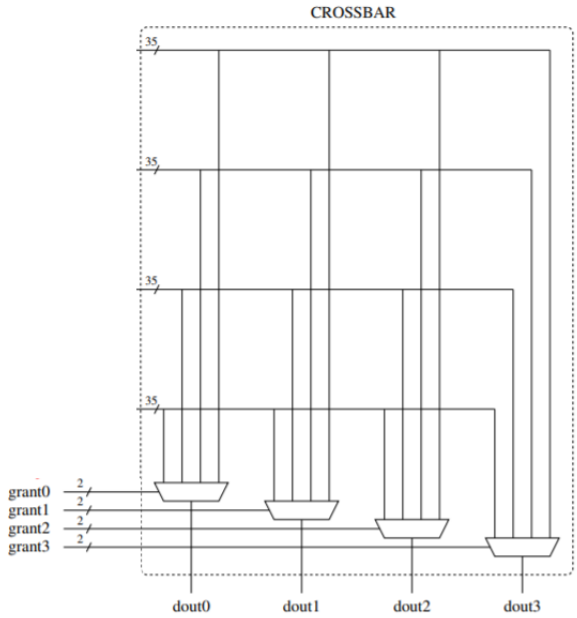


RD_block allows to read from fifo when for requested output(req_i_t) we get value identical to certain block, which are (00,01,10,11).

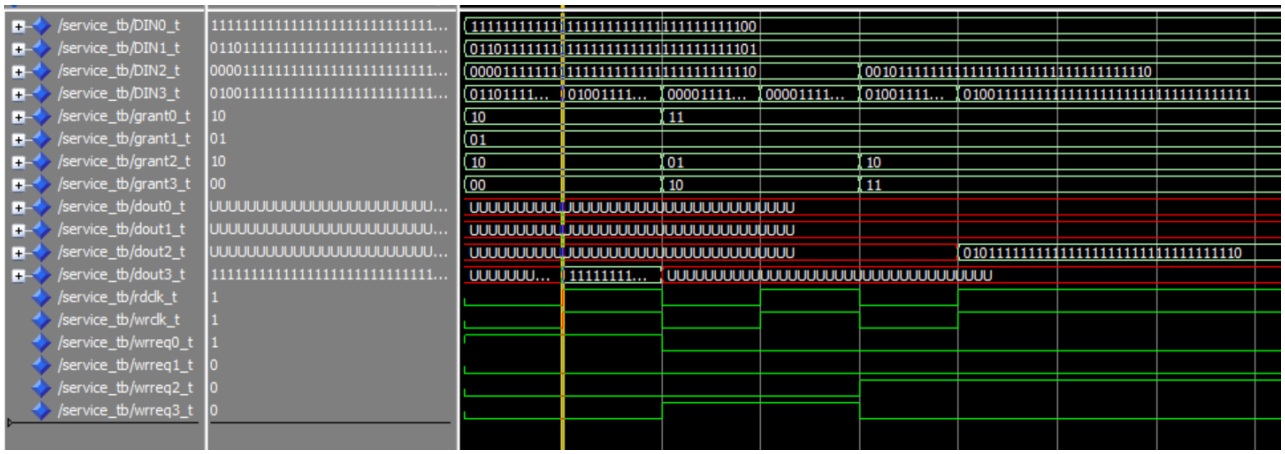
Crossbar:



Group of four multiplexer, each has the same data for inputs and various selector signals.



Complete circuit contains: RD_BLOCK, CROSSBAR, MUXs, My_FIFO:



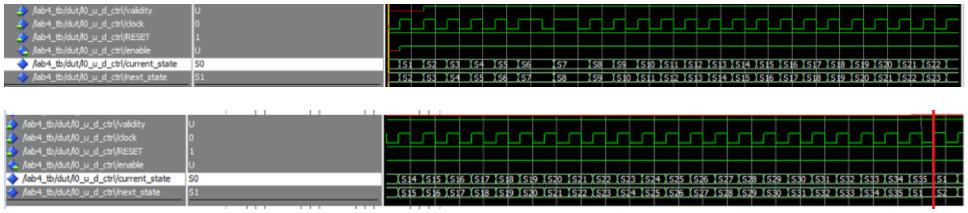
Undefined output indicates, that READ_signal has not been generated. According to this grant signals detemine if input is accepted or no. For example, DIN0_t requests output third (”11”), for grant3_t we haev ”00”. So in this situation RD_Block is able to produce logical ’1’, that is why on output 3 we can see DIN0_t.

Lab.3

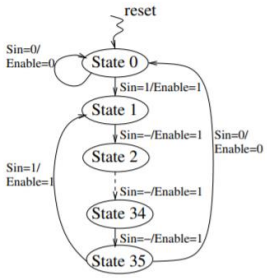
3.2 Control Unit of SER/DES units

3.2.1

Deserializer FSM:

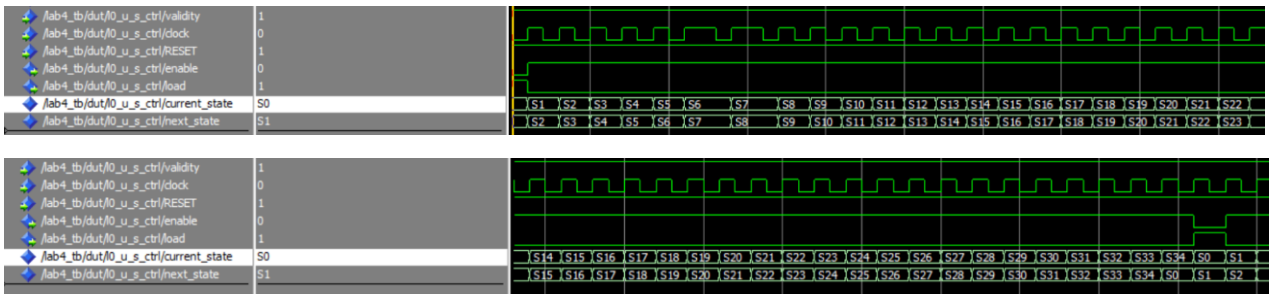


It works according to the graph:

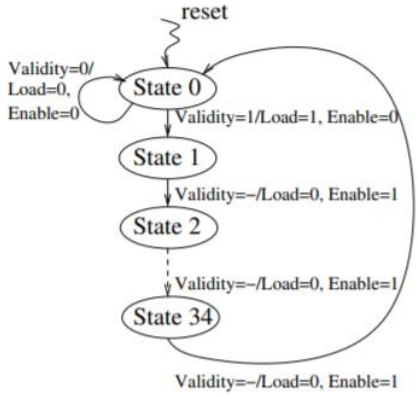


3.2.1

Serializer FSM:



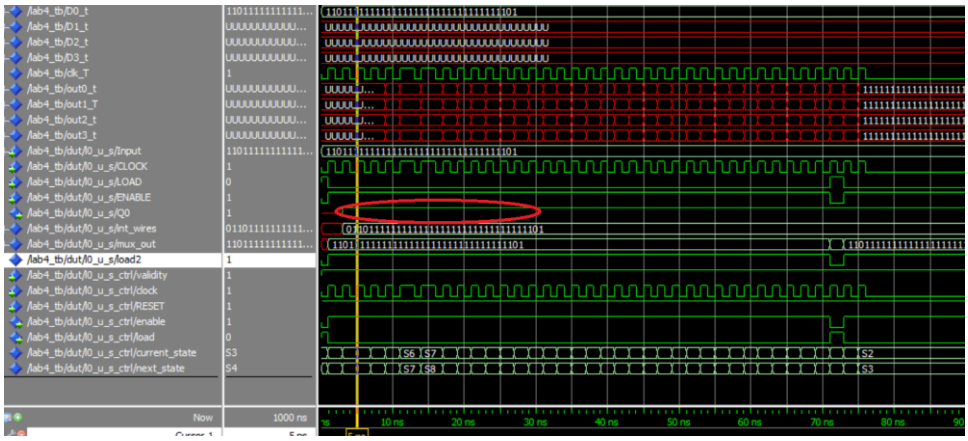
It refers to:



SER FSM

Lab.4

I got problem with Q0. It should change but it is stable, all time value is one.



In conlusion, I did all three labs and I tried also the fourth, but I did not complete it.

Despite this fact, I learn a lot how to describe hardware behaviour and how to test it or evaluate the work.

Wojciech Paluszkiewicz, s293958