FPGA RAM (C1)

Young Won Lim
5/13/16

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using LibreOffice and Octave.

Virtex LUT RAM

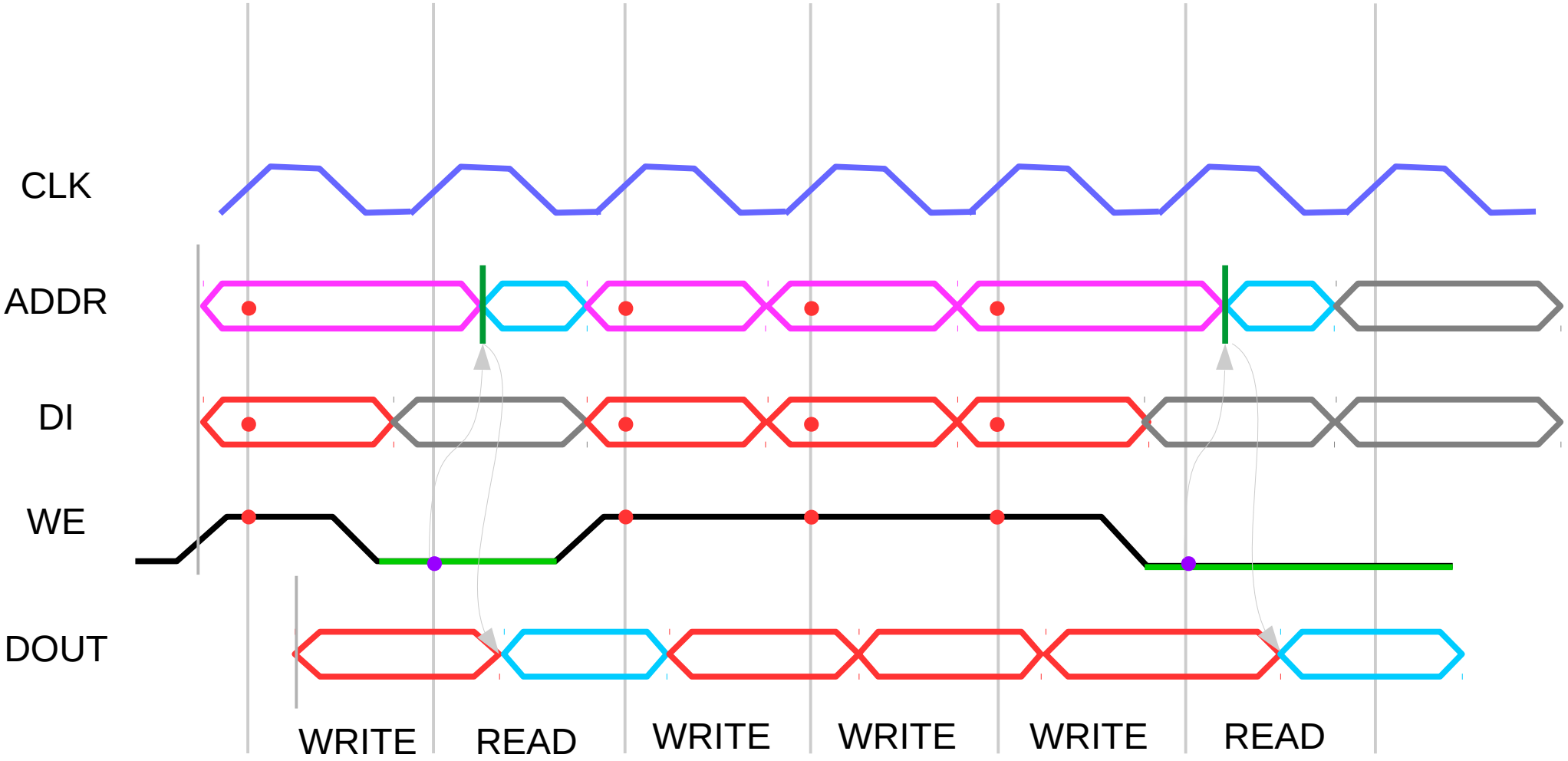
```
module ram16x1(q, a, d, we, clk);
output      q;
input       d;
input [3:0] a;
input       clk, we;

reg         mem [15:0];

always @(posedge clk) begin
   if(we)
      mem[a] <= d;
end

assign q = mem[a];

endmodule
```

Synchronous Write

Asynchronous Read

http://www-inst.eecs.berkeley.edu/~cs150

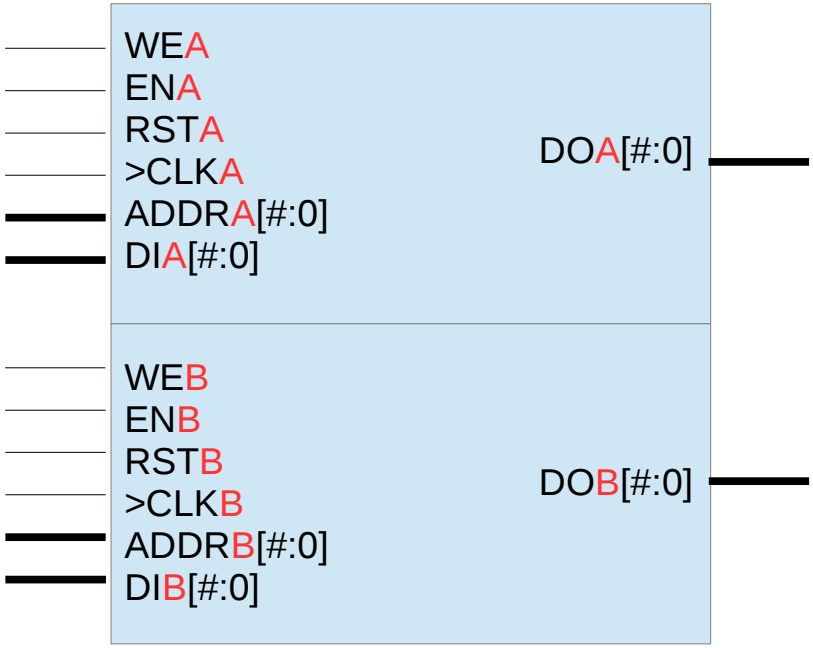
LUT RAM Timing



Xilinx Document

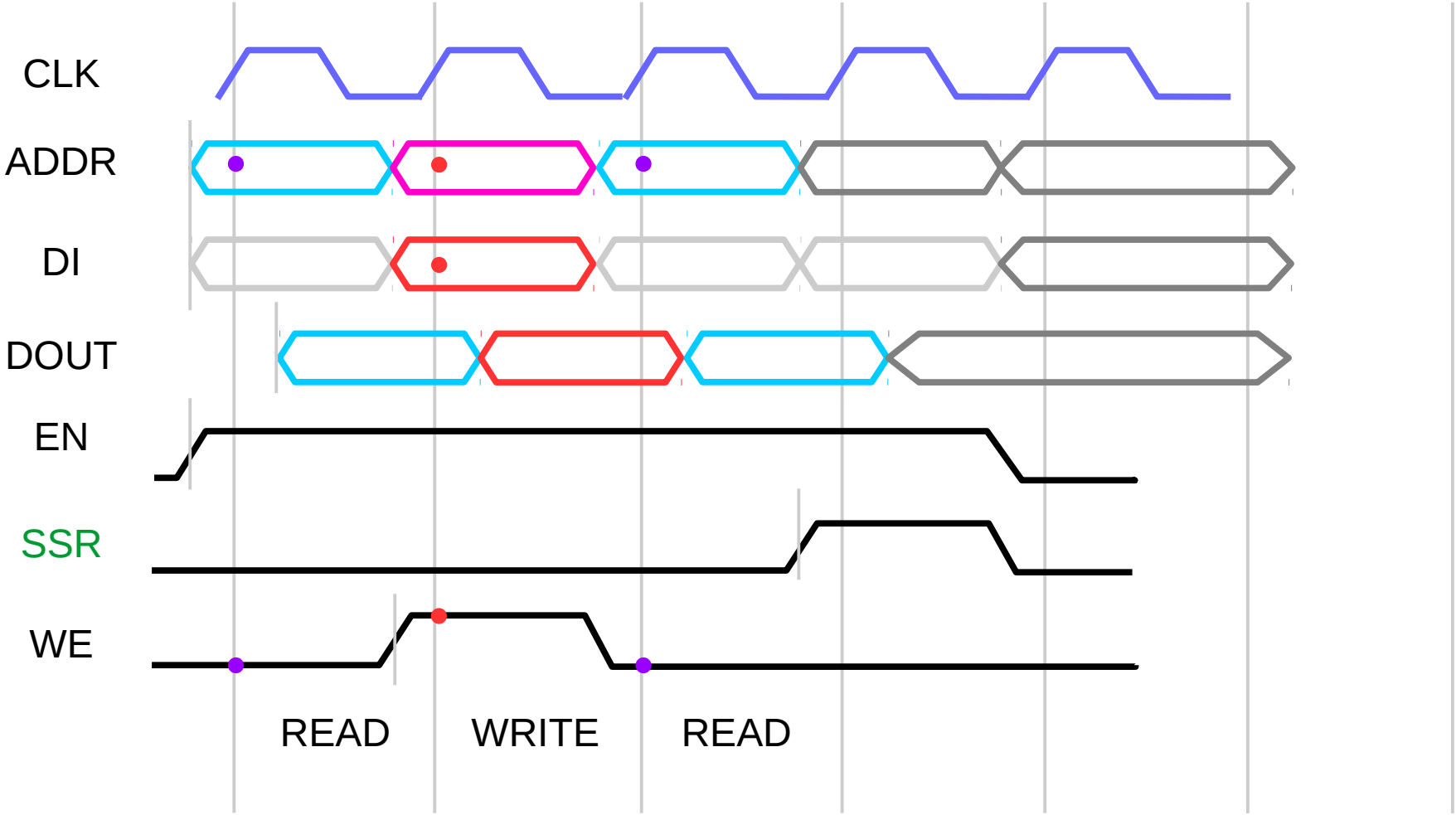
Virtex Block RAM

Every block SelectRAM
- synchronous write
- syncrhronous read
- dual-ported
- independent control signals
- independent data widths
- independent CLKA & CLKB
- 4096-bit RAM



http://www-inst.eecs.berkeley.edu/~cs150

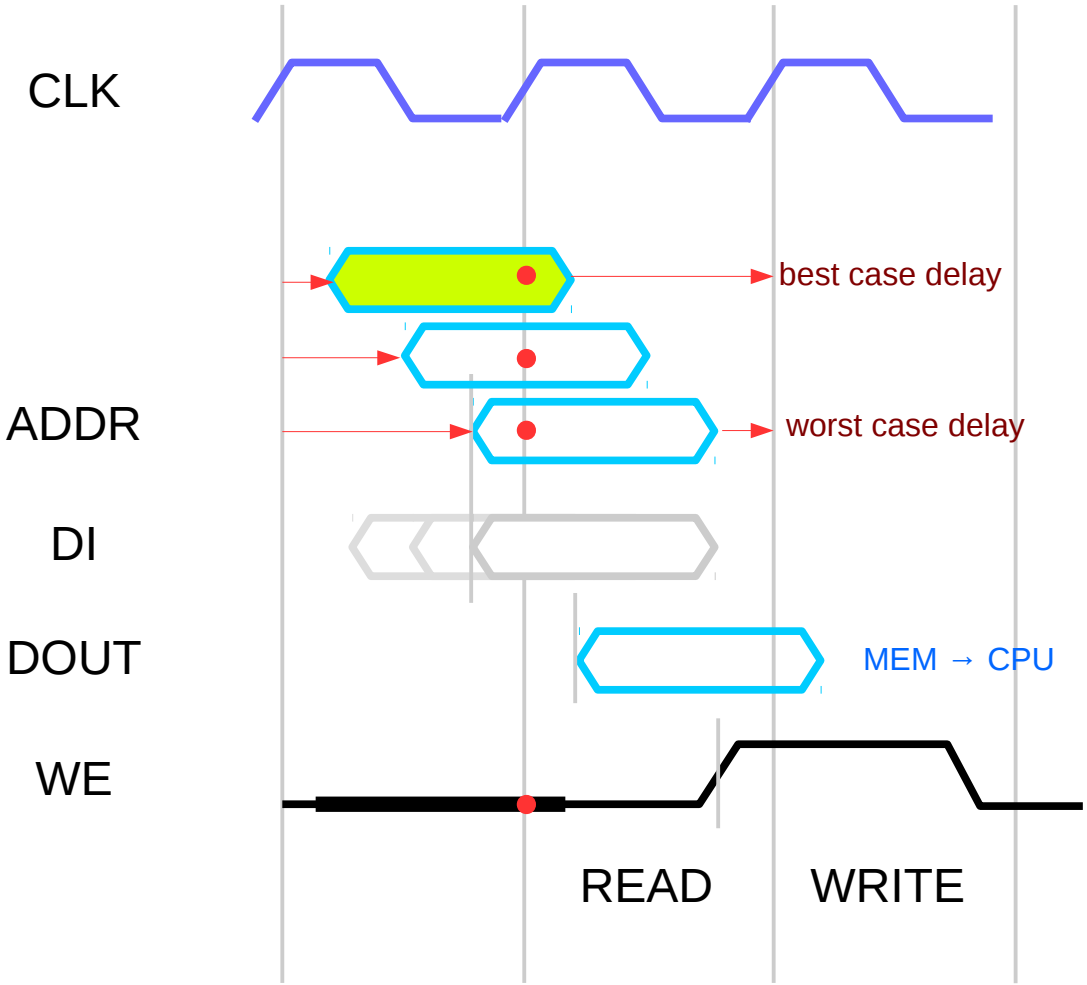
Virtex Block RAM Timing



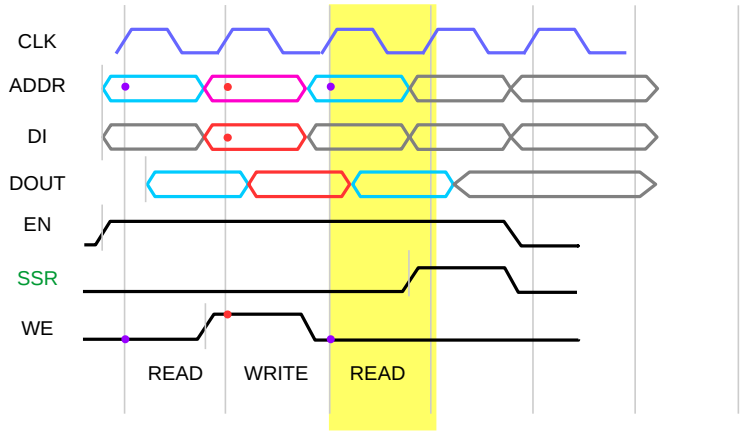
Xilinx Document

SSR (Synchronous Set Rest) value = 1010

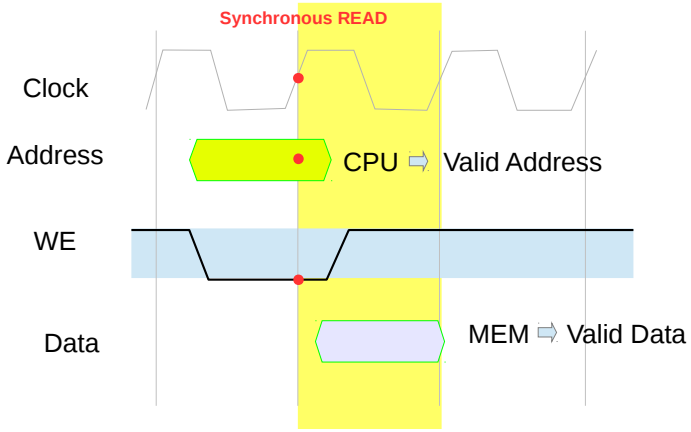
Sync **RD** Cycle - worst & best case



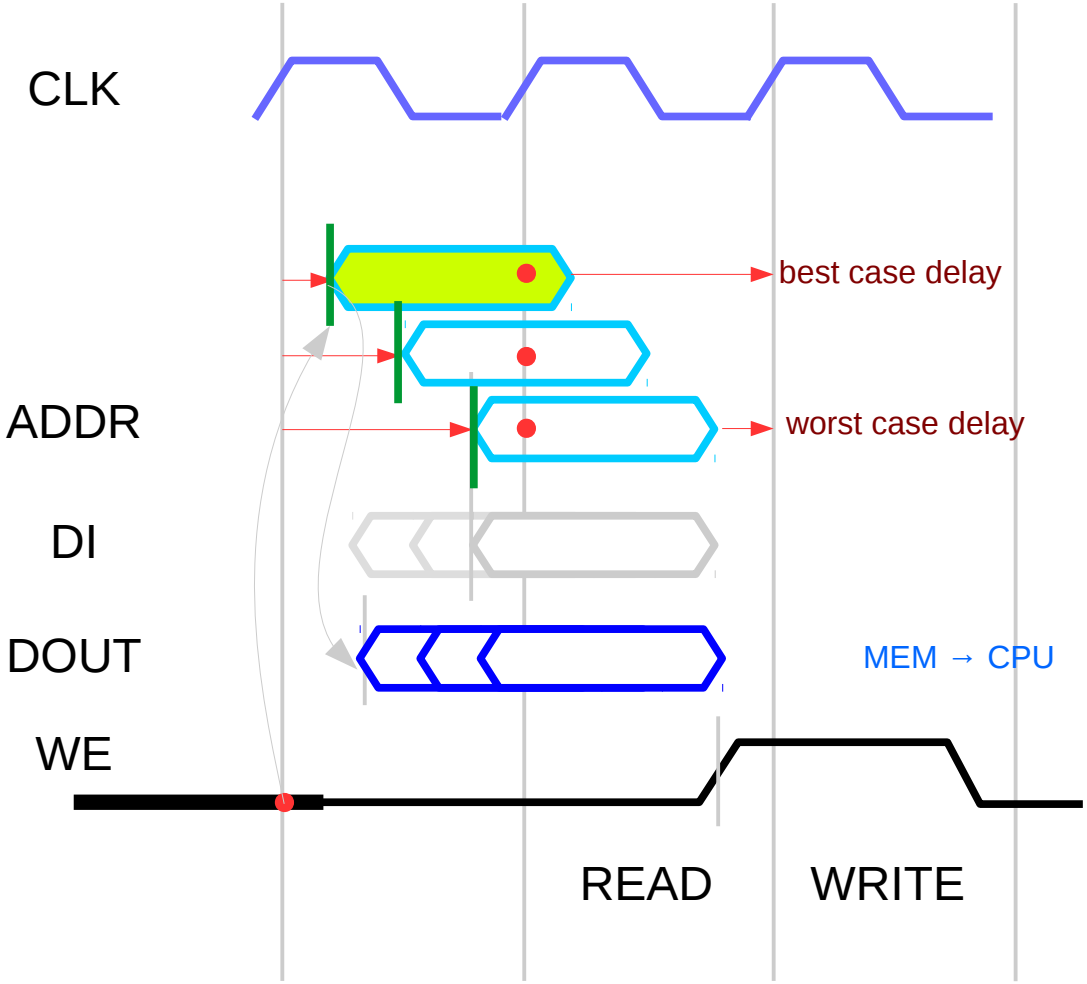
Timing Specification



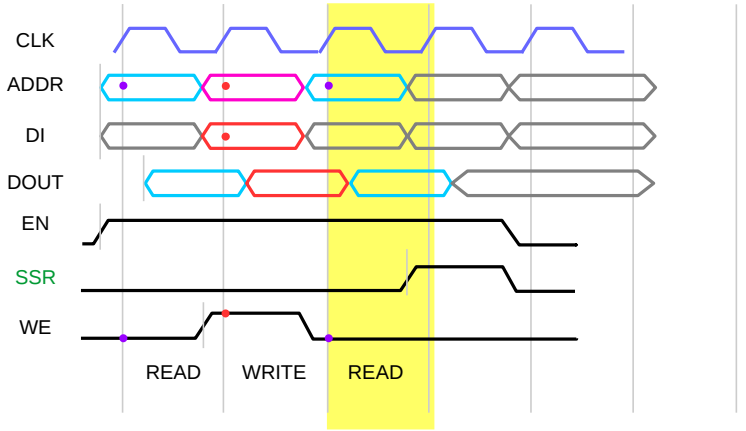
more realistic timing



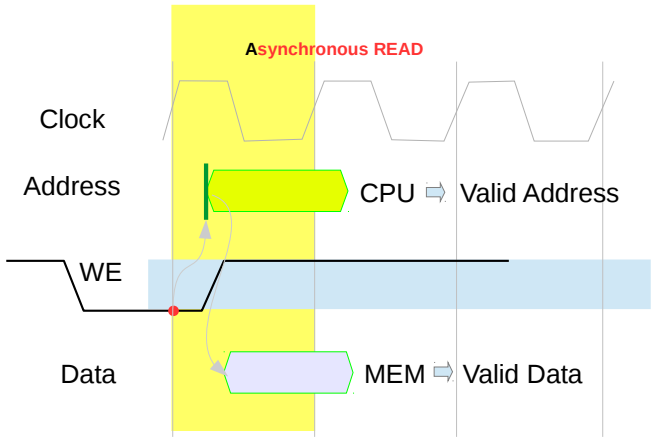
Async **RD** Cycle - worst & best case



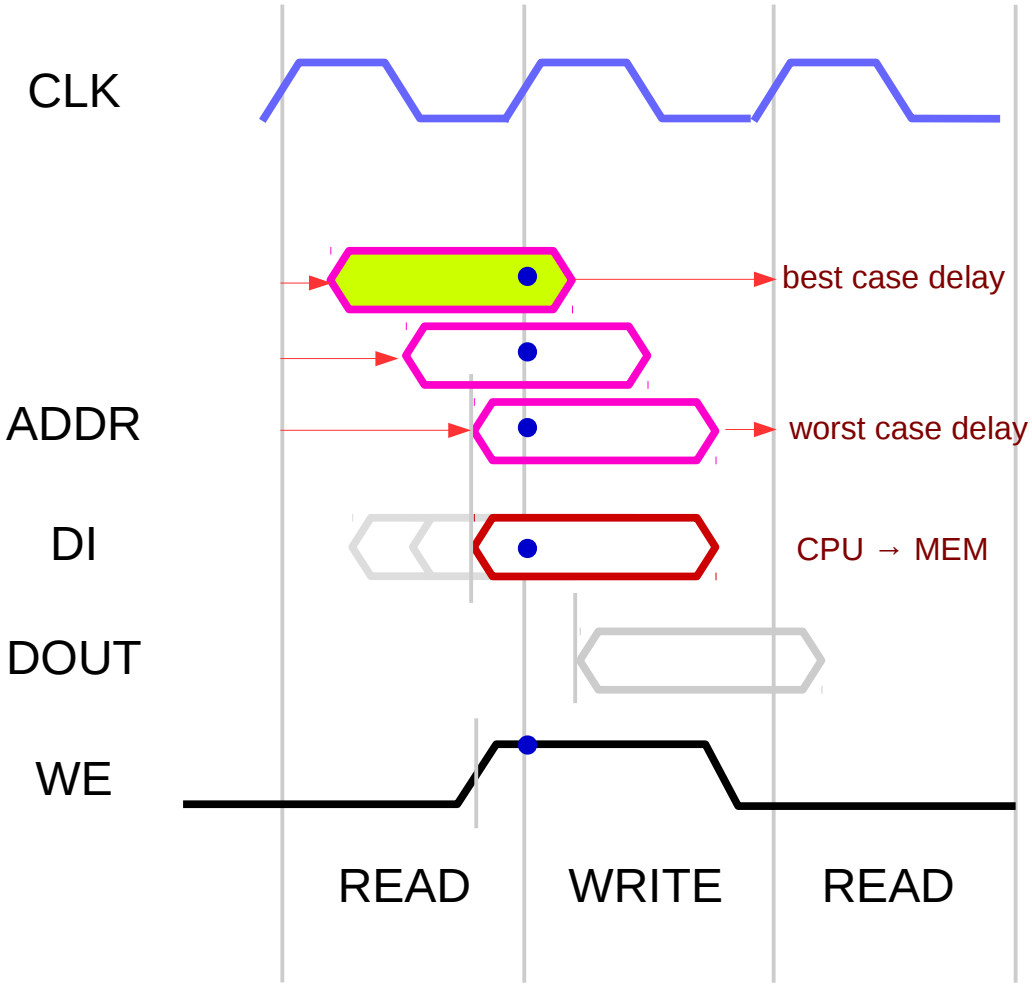
Timing Specification

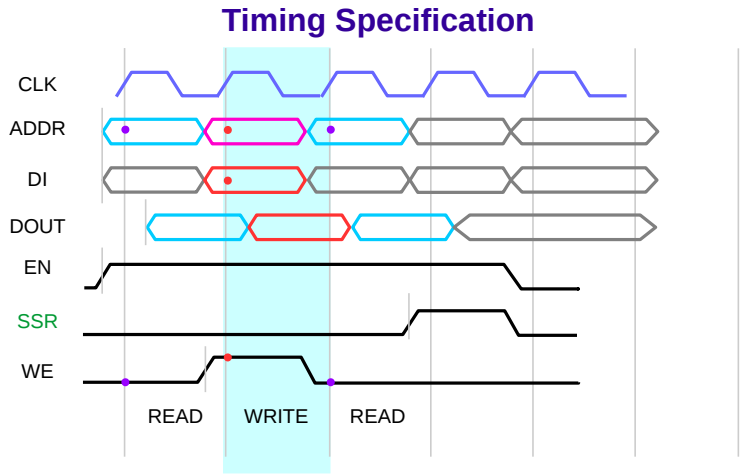


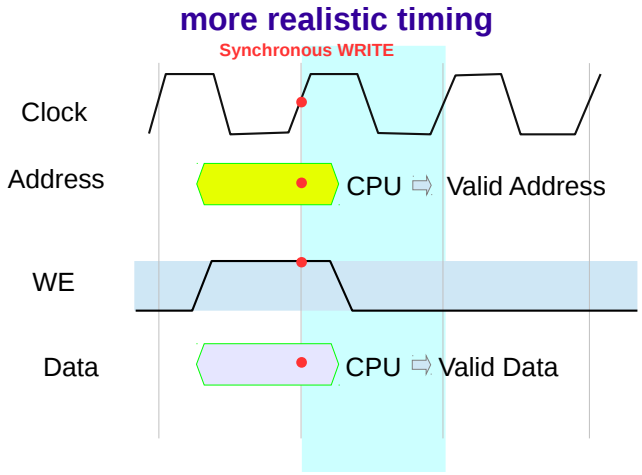
more realistic timing



WR Cycle – worst & best case







Waveform Viewer Timing (1)

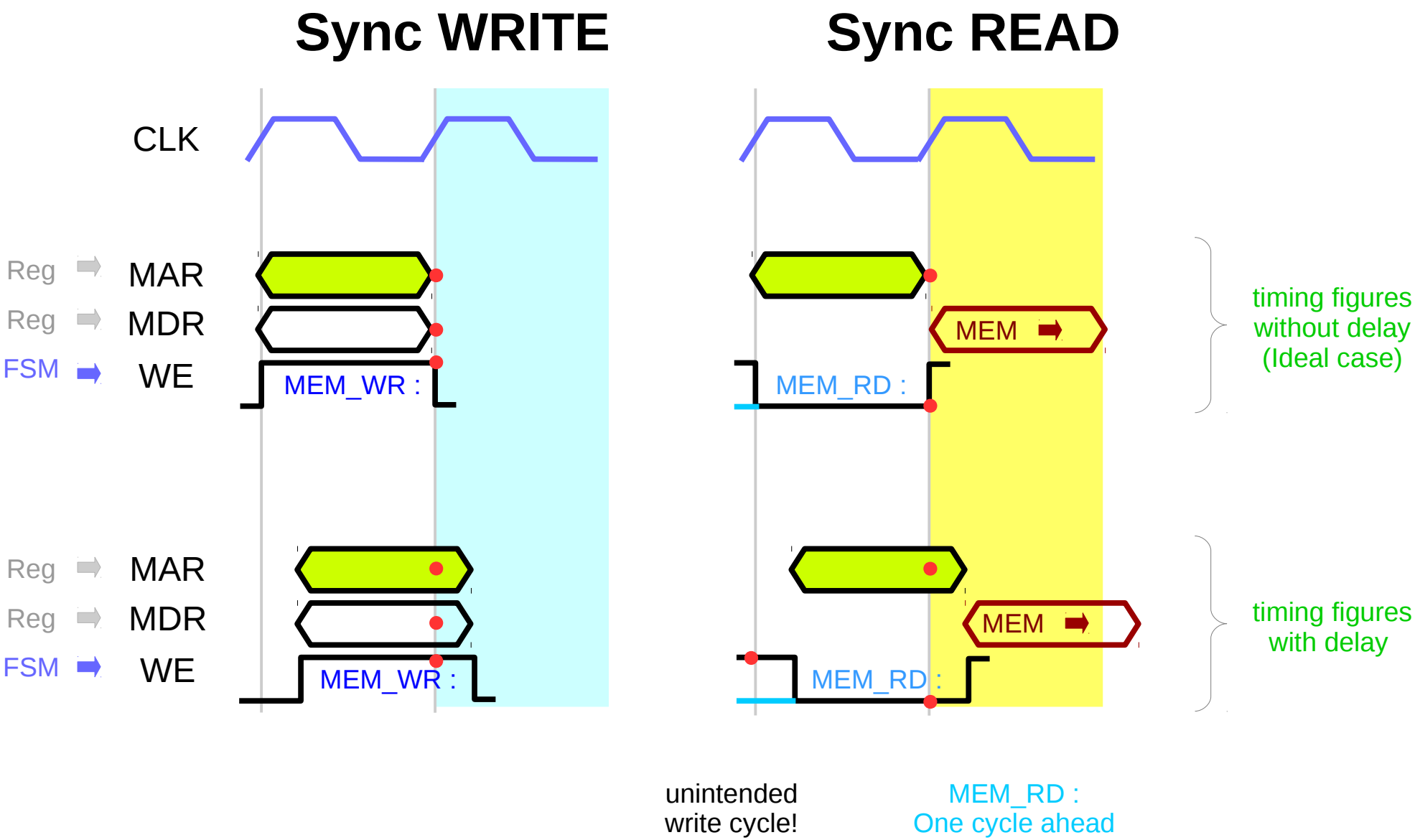
* timing figures without delay (Ideal case)

RTL functional simulation

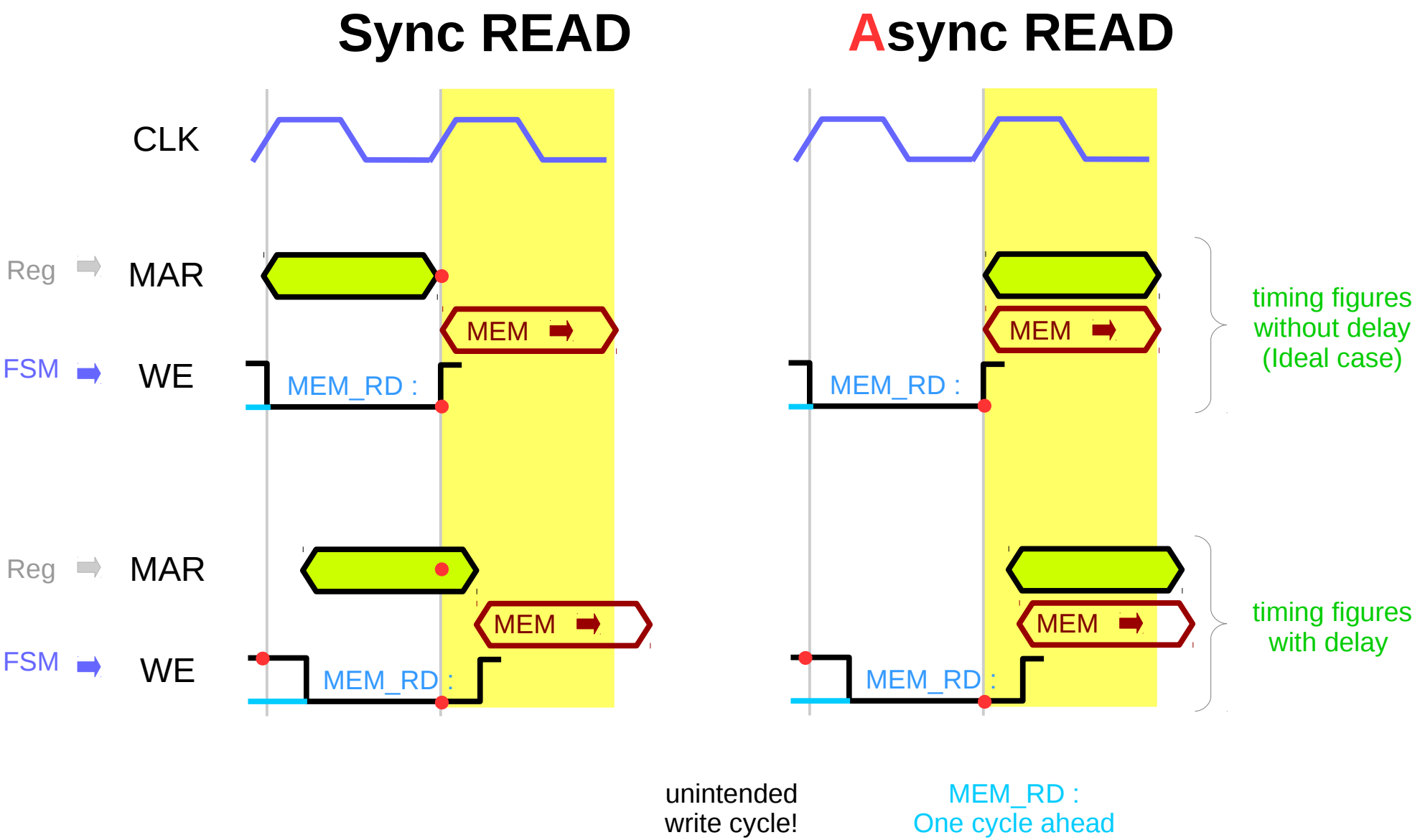
* timing figures with delay

Gate level simulation
with SDF annotation

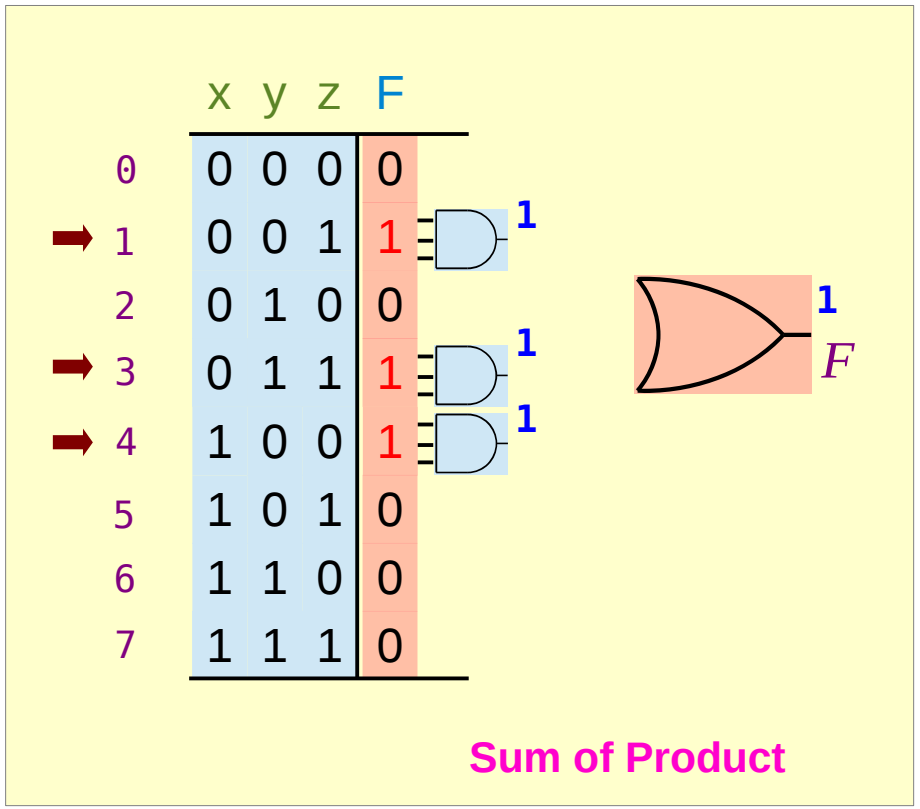
Waveform Viewer Timing (2)

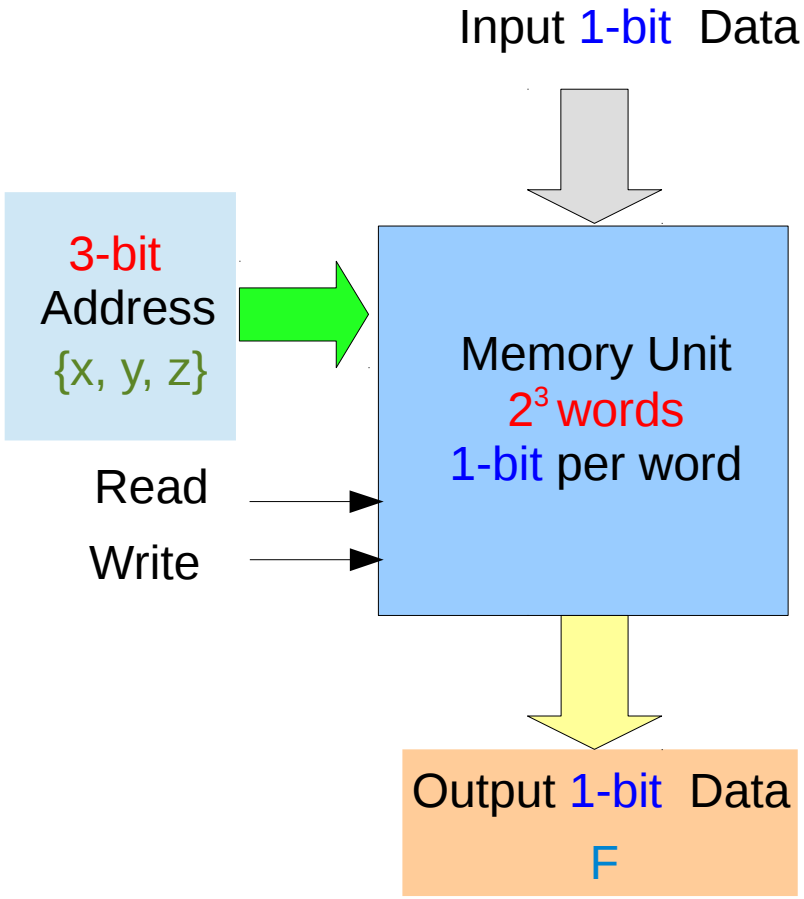


Waveform Viewer Timing (3)



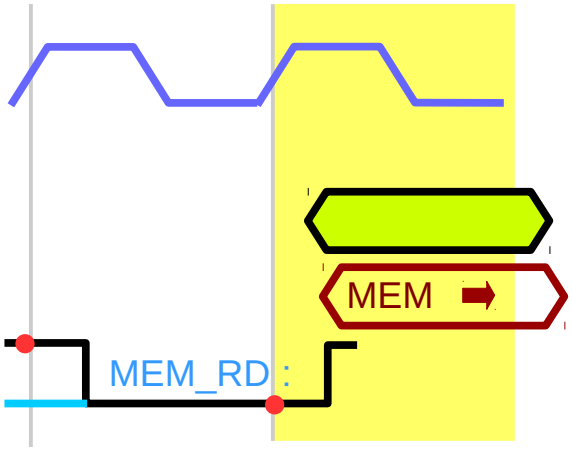
LUT as a combination logic block

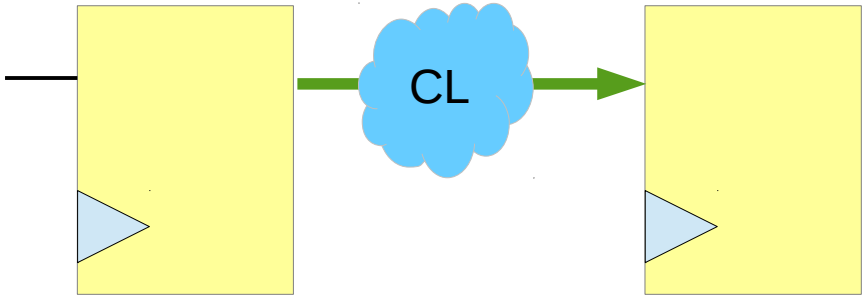




LUT : Async Read

Async READ



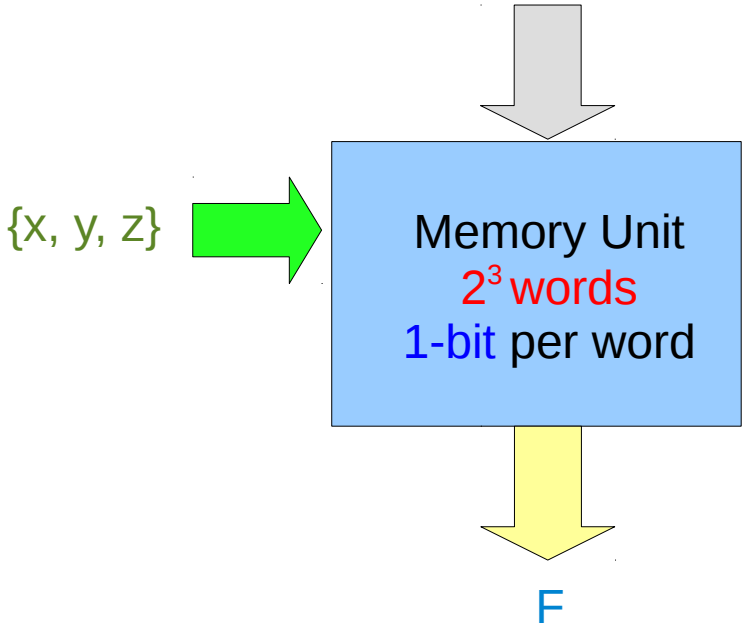


Address : Inputs
Data : Outputs

The outputs available in the same cycle where the inputs are applied

Combinational Logic Block :
A set of Boolean Functions

LUT Data Write



References

[1] http://en.wikipedia.org/
[2] https://en.wikiversity.org/wiki/The_necessities_in_SOC_Design
[3] https://en.wikiversity.org/wiki/The_necessities_in_Digital_Design
[4] https://en.wikiversity.org/wiki/The_necessities_in_Computer_Design
[5] https://en.wikiversity.org/wiki/The_necessities_in_Computer_Architecture
[6] https://en.wikiversity.org/wiki/The_necessities_in_Computer_Organization
[7] https://en.wikiversity.org/wiki/Understanding_Embedded_Software
[8] Digital Systems, Hill, Peterson, 1987
[9] http://en.wikipedia.org/
[10] http://www.ele.uri.edu/Courses/ele306/f01/Tinydoc.pdf