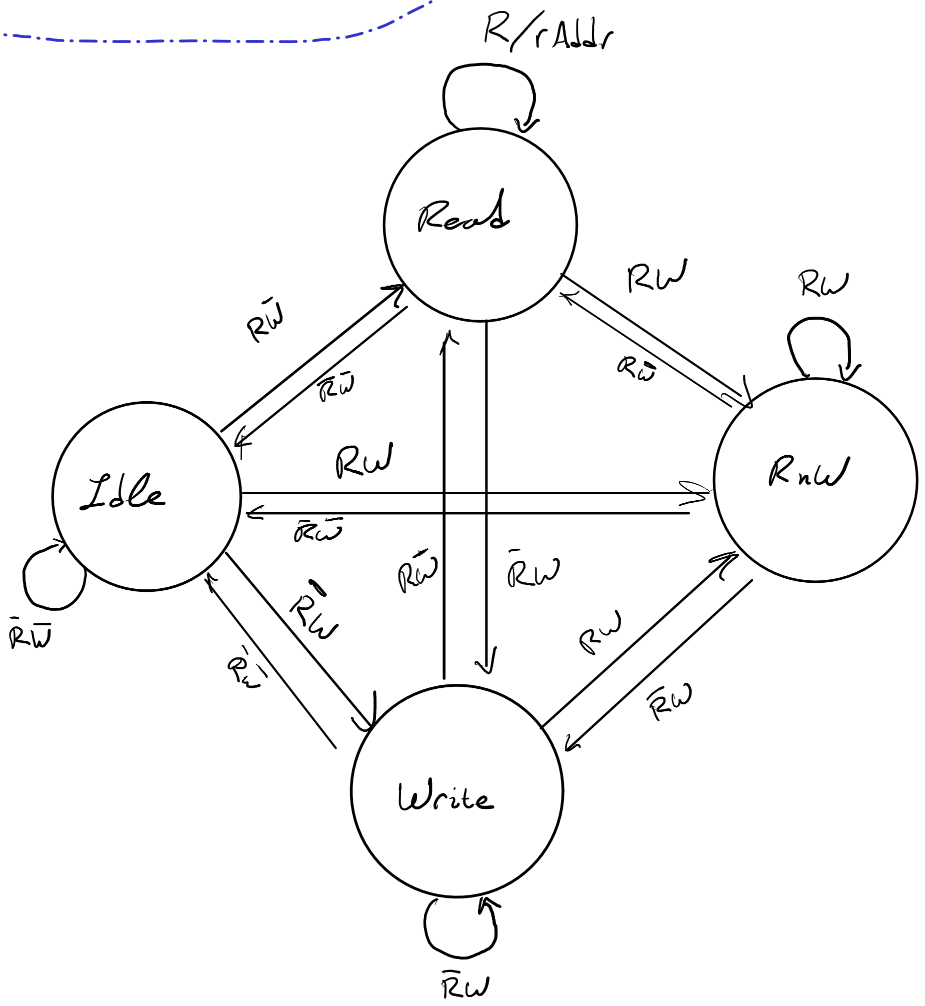
States

Write
Read
Read & Write
Idle
Initial state



For RW conflict:

1) Read data from RAddr
2) Write data to WAddr

This way even if WAddr = Raddr there will be no race condition

Idle state:

In this state the output of the NumOUT will be the last output but the Valid flag will be 0

Note: Even in the Write state the NumOUT will be an Unknown or the last value but the Valid flag will be off

Memory Block (Single Port RAM)

Input

we

din

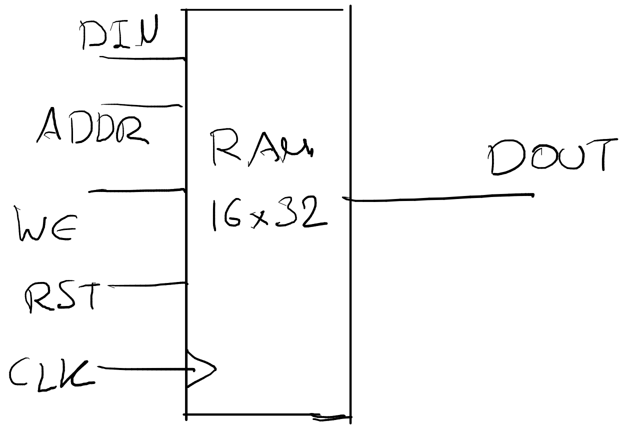
addr

rst

clk

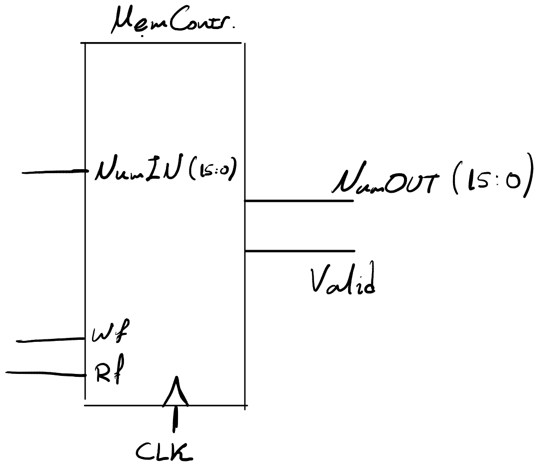
Out

dout



Memory Controller

For the most part it is the implementation of the FSM machine that we have analyzed (NOT)



Read

- Rf must be 1
- Set the addr to the memory from Raddr
- Set WE flag to 0
- Redirect RAM Dout to NumOUT
- Set Valid to 1

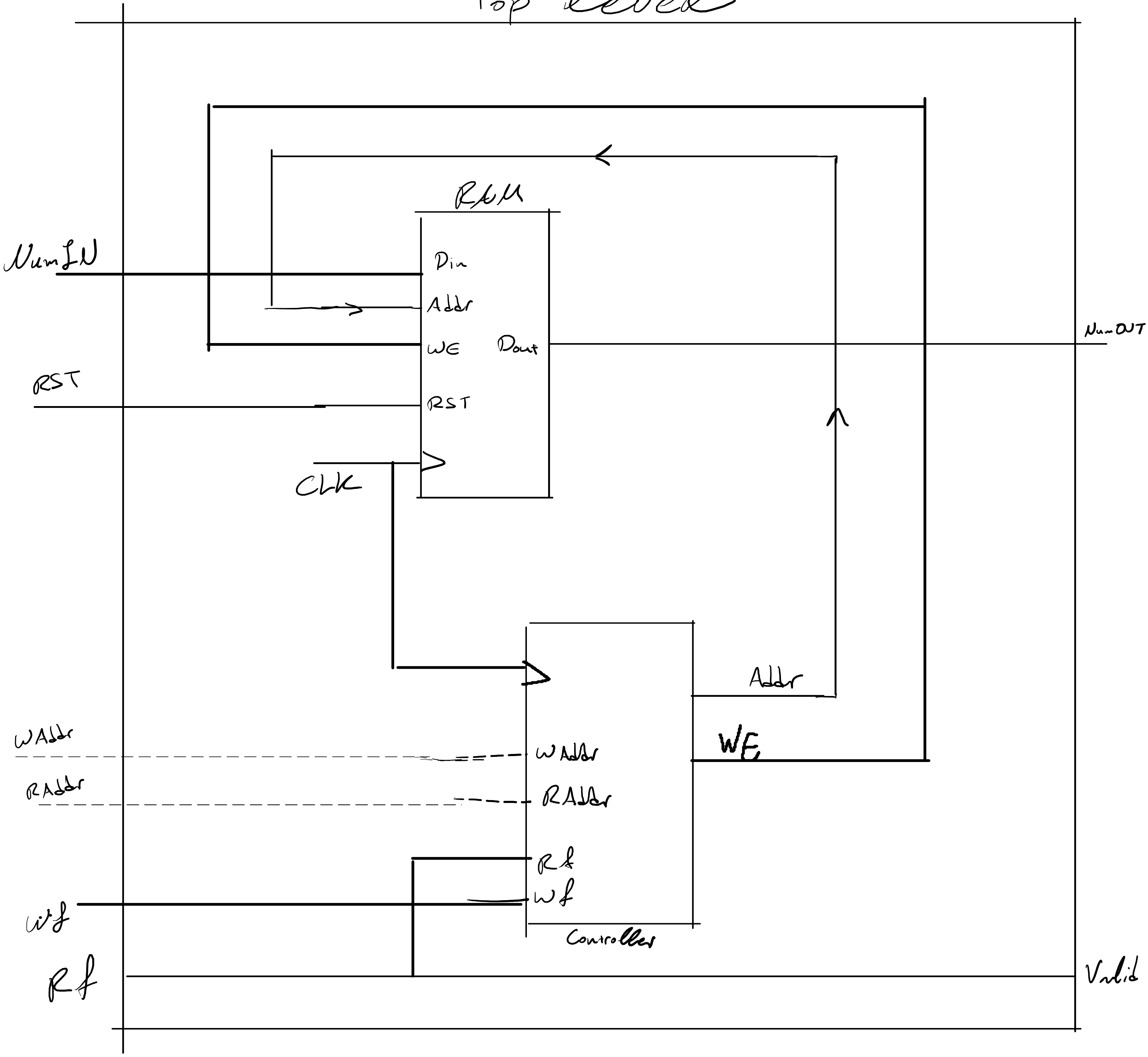
Write

- Rf = 0, Wf = 1
- Valid = 0
- Set Mem addr to Waddr value and NumIn to RAM Din
- Set NumOUT to RAM Dout

Read & Write

- Rf = 1 & Wf = 1 & Valid = 1
- Mem Addr = RAddr
- Output Read Value
- MemAddr = WAddr
- Write NumIN to RAM

Top level


RAM
NumIN
Din
Addr
WE
Dout
NumOUT
RST
RST
CLK
Addr
WE
WAddr
WAddr
RAddr
RAddr
Rf
Wf
Wf
Controller
Rf
Valid