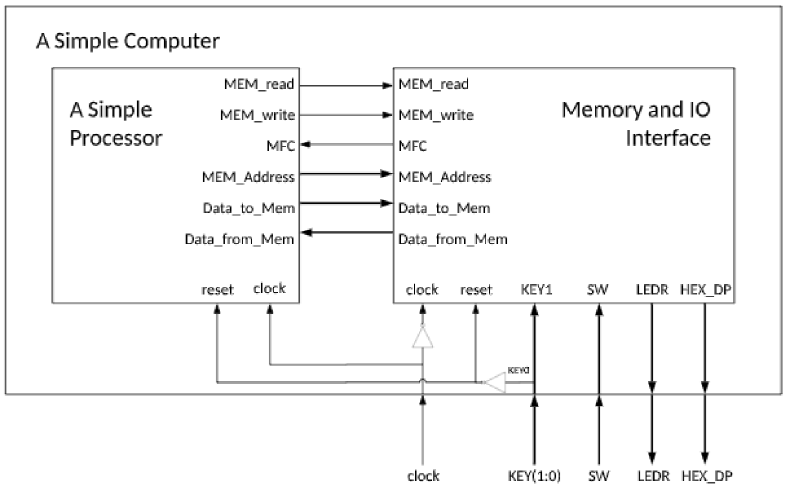
CSCE 230 – Group Labs (the Project)

5. Computer organization

5.1. A simple computer consists of a simple processor and a memory and IO interface.



5.2. VHDL designs

- A Simple Computer: SimpleComputer.vhd, which is complete and is the **top-level entity** for Quartus.
- A Simple Processor: SimpleProcessor.vhd. which is **incomplete** and is what you need to complete in this project.
- Memory and I/O:
  - MemoryIOInterface.vhd, which is complete.
  - MainMemory.vhd, which is complete.
  - MemoryInitialization.mif, which is **incomplete** and is where you need to write the binary/hexadecimal encodings of an assembly program. This mif file will be automatically read by MainMemory.vhd to initialize the initial values of memory.

5.3. Computer Interface

- The processor clock input port is directly connected to the computer clock input port, and is triggered by a rising edge of the computer clock. The memory and IO clock input port is connected to the output of a NOT gate, and thus is triggered by a falling edge of the computer clock.
- When KEY0 is pushed (i.e. KEY0=0), the computer is reset. Therefore, the inverted KEY0 is connected to the reset input ports of the processor and memory and IO interface.

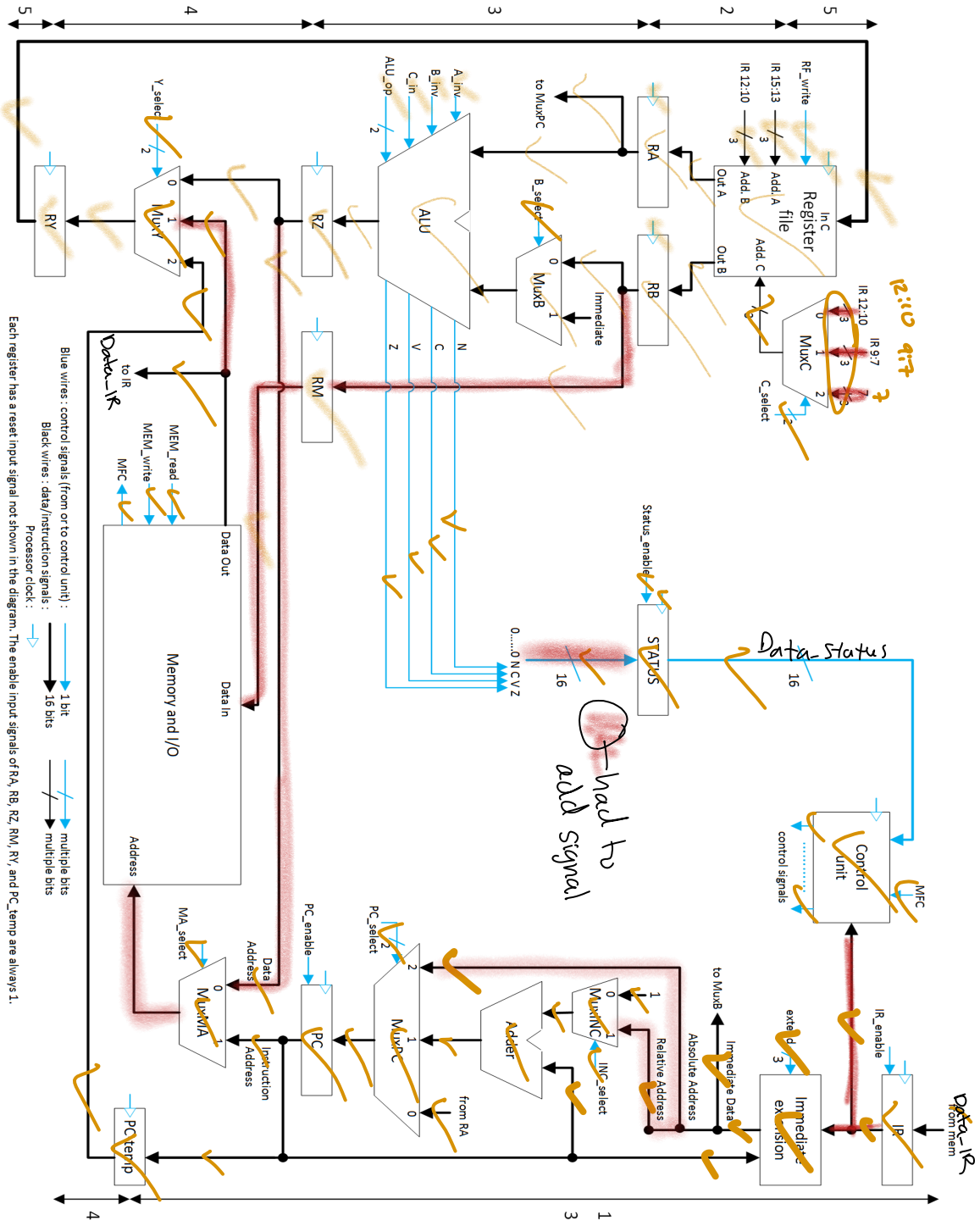
5.4. Main memory and I/O ports

Base Address	End Address	Memory or I/O	16-bit data at an address
0x0000	0x0FFF	Main memory	
0x1000	0x1000	LEDR: LED Red (Write only)	000000:LEDR9:LEDR8:...:LEDR0
0x1010	0x1010	HEX_DP: Hex Seg (Write only)	00000000:HEX57:HEX47:...:HEX07
0x1040	0x1040	SW: Slider Switch (Read only)	000000:SW9:SW8:...:SW0
0x1050	0x1050	KEY: Push Button (Read only)	000000000000:KEY1:0

CSCE 230 – Group Labs (the Project)

6. Processor architecture

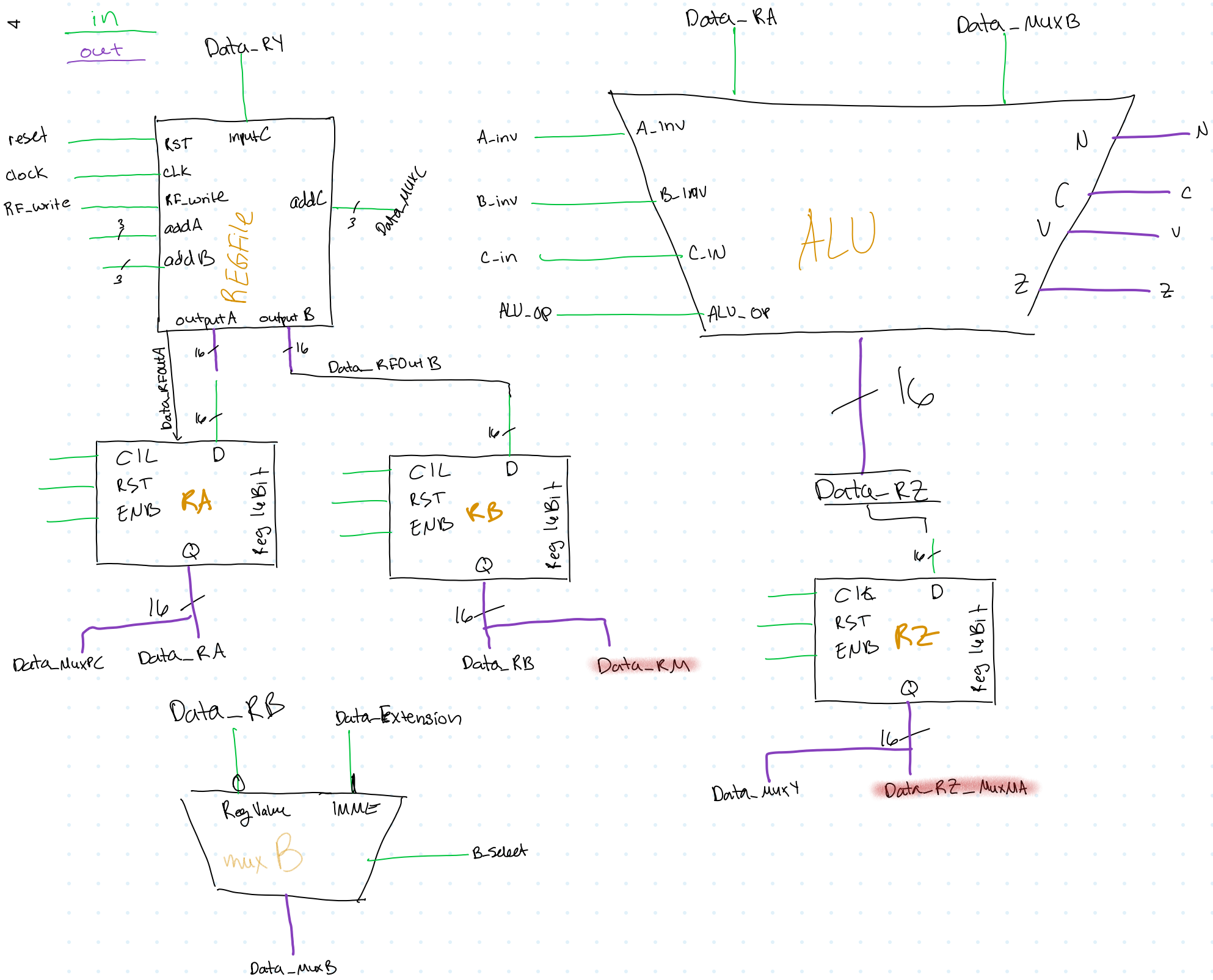
6.1. The processor architecture diagram

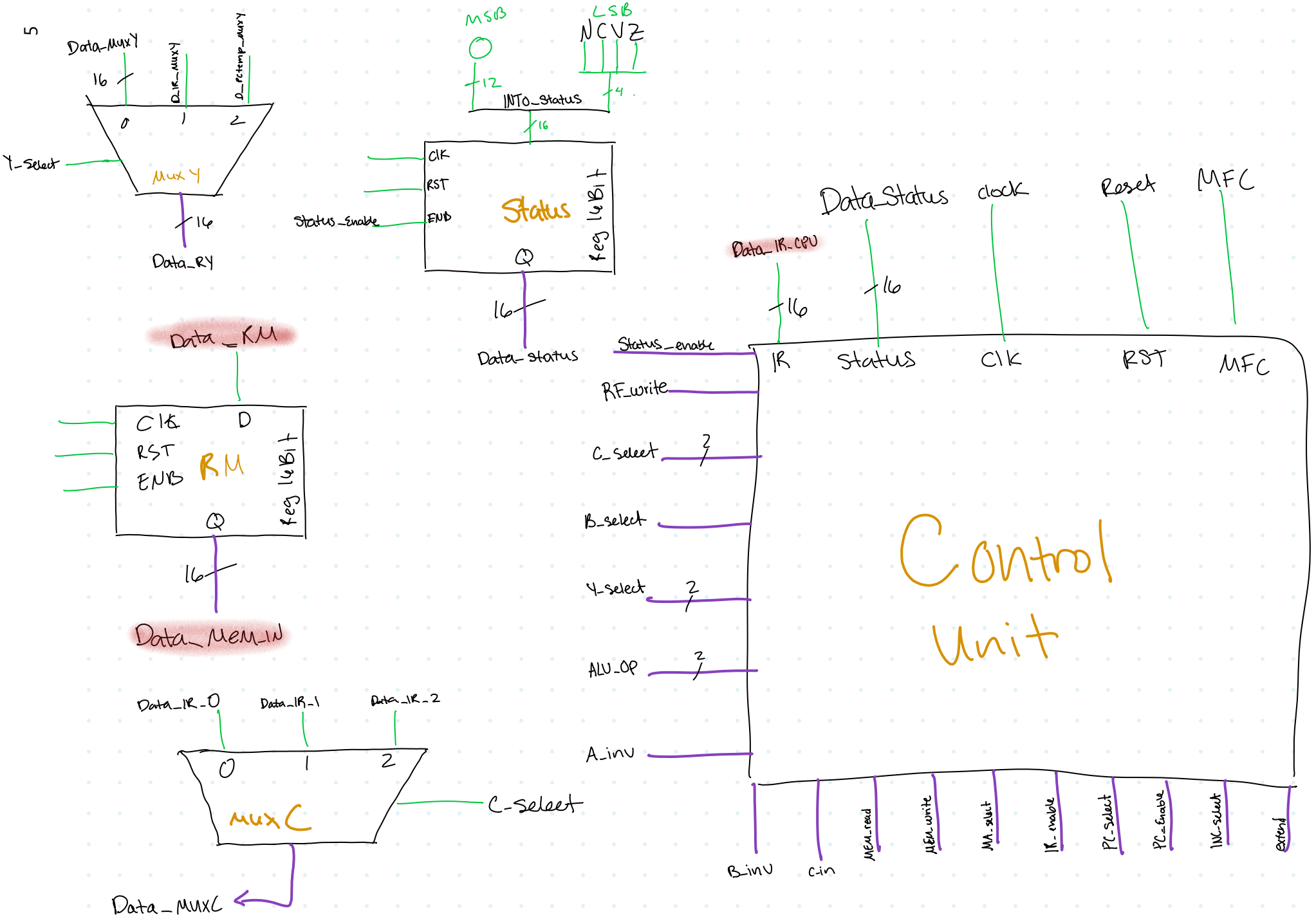


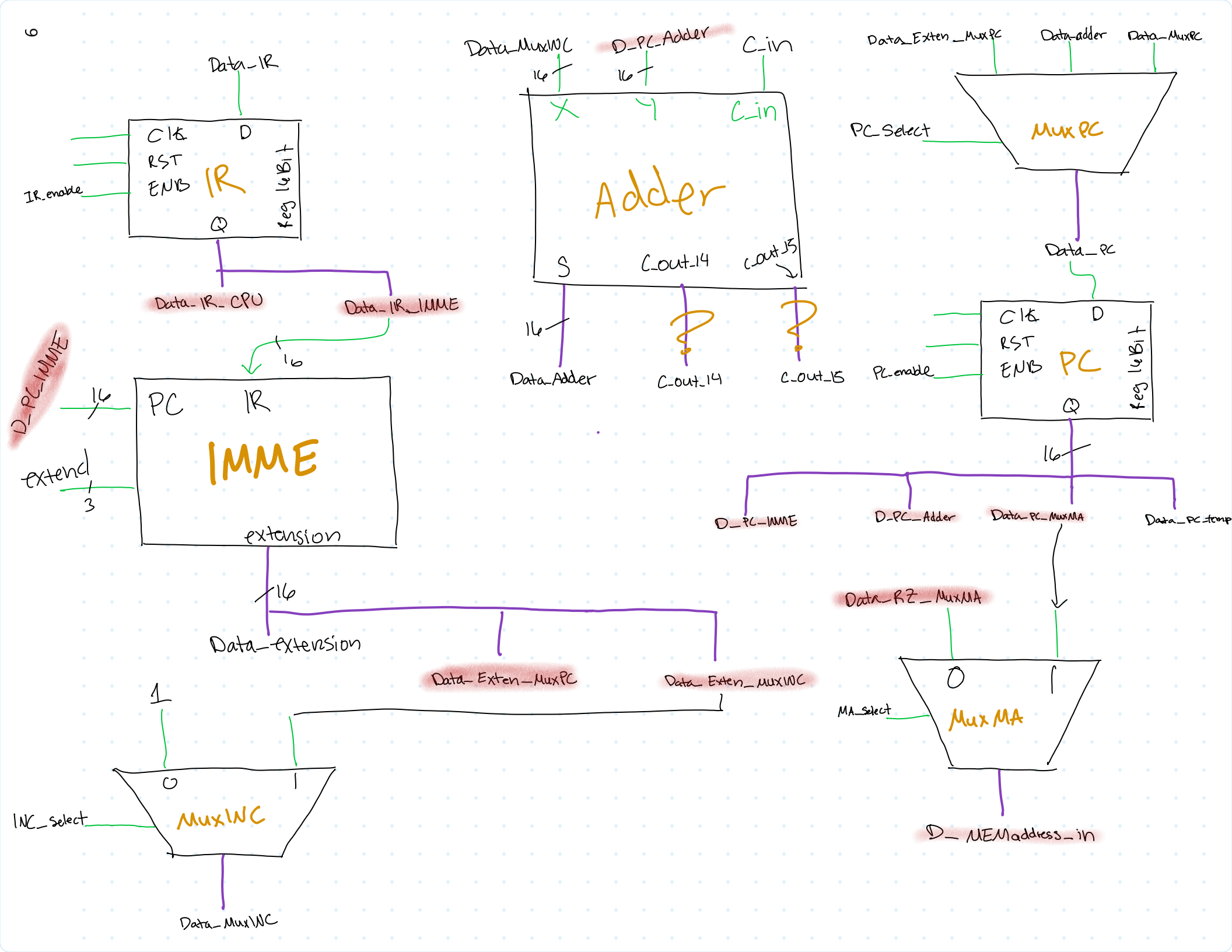
4

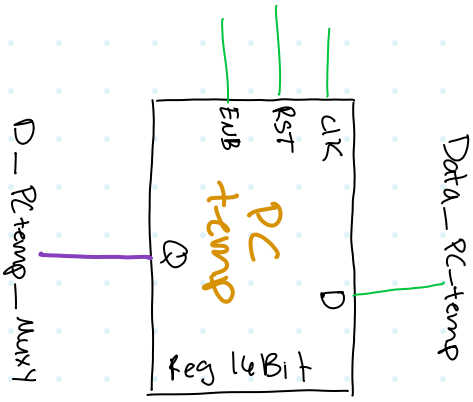
in

out


Data_RY
reset
clock
RF_write
RST
CLK
RF_write
addA
addB
inputC
addC
REGFile
outputA
outputB
3
3
3
Data_MuxC
16
16
Data_RFOutA
Data_RFOutB
16
16
CIL
RST
ENB
D
RA
Reg 16Bit
Q
16
Data_MuxPC
Data_RA
CIL
RST
ENB
D
RB
Reg 16Bit
Q
16
Data_RB
Data_RM
Data_RA
Data_MuxB
A_inv
B_inv
C_in
ALU_OP
A_inv
B_inv
C_IN
ALU_OP
ALU
N
C
V
Z
N
C
V
Z
16
Data_RZ
16
CIK
RST
ENB
D
RZ
Reg 16Bit
Q
16
Data_MuxY
Data_RZ_MuxMA
Data_RB
Data_Extension
0
1
RegValue
IMME
muxB
B_select
Data_MuxB


Data_MuxY
16
D_IR_muxY
D_pctemp_muxY
0
1
2
Y_select
MuxY
16
Data_RY
MSB
LSB
NCVZ
12
INTO_Status
4
16
CLK
RST
ENB
Status
Reg 16Bit
Status_Enable
Q
16
Data_status
Data_RM
CLK
D
RST
ENB
RM
Reg 16Bit
Q
16
Data_MEM_IN
Data_IR_0
Data_IR_1
Data_IR_2
0
1
2
MuxC
C_select
Data_MUXC
Data_IR_CPU
16
Data_Status
16
clock
Reset
MFC
Status_enable
IR
Status
CLK
RST
MFC
RF_write
C_select
2
B_select
Y_select
2
ALU_OP
2
A_inv
Control
Unit
B_inv
C_in
MEM_read
MEM_write
MA_select
IR_enable
PC_select
PC_Enable
INC_select
extend


Data_IR
Clk
RST
ENB
IR
D
Q
Reg 16Bit
IR_enable
Data_IR_CPU
Data_IR_IMME
16
D_PC_IMME
16
PC
IR
IMME
extend
3
extension
16
Data_extension
Data_Muxinc
D_PC_Adder
C_in
16
16
X
Y
C_in
Adder
S
C_out_14
C_out_15
16
Data_Adder
C_out_14
C_out_15
Data_Exten_MuxPC
Data_adder
Data_MuxPC
PC_Select
MuxPC
Data_PC
Clk
RST
ENB
PC
D
Q
Reg 16Bit
PC_enable
16
D_PC_IMME
D_PC_Adder
Data_PC_MuxMA
Data_PC_temp
Data_RZ_MuxMA
Data_Exten_MuxPC
Data_Exten_MuxINC
1
0
1
MuxINC
INC_select
Data_MuxINC
0
1
MuxMA
MA_select
D_MEMaddress_in


EnB
RST
CLK
PC temp
D
Q
Reg 16Bit
D_PC temp_Muxy
Data_PC_temp