Control Signal:

- WB
- Read_Mem
- Write_Mem
- immediate_value(Mux before ALU)
- branch_pc (MUX before PC)
- branch (signal from control unit)
- ALU_selection_lines (4 bits temp)
- Port flag (for input port)
- SP (is addressing going into data memory stack pointer or normal memory address)
- Return (1 for RET & RETI, 0 For other instruction)
- Push
- Pop

Instruction	RTI	Int	Men_to_Reg	WB	Read_Mem	write_MEM	immediate_value	No_Cond_Branch	branch	SP	Port.flag	Return	Push	Pop	Call	ALU_selection_lines
Interrupt	0	1	0	0	0	1	0	0	0	1	0	0	1	0	0	1010
NOP	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1010
SETC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0100
CLRC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0110
OUT Rsrc1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1110
IN Rdst	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1010
								R-Types								
INC Rdst, Rsrc1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1100
DEC Rdst,	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1101

Instruction	RTI	Int	Men_to_Reg	WB	Read_Mem	write_MEM	immediate_value	No_Cond_Branch	branch	SP	Port.flag	Return	Push	Pop	Call	ALU_selection_lines
Rsrc1																
NOT Rdst, Rsrc1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1111
MOV Rdst, Rsrc1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1110
ADD Rdst, Rsrc1, Rsrc2	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0111
IADD Rdst, Rsrc1, Imm	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0111
SUB Rdst, Rsrc1, Rsc2	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0001
AND Rdst, Rsrc1, Rsrc2	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0010
OR Rdst, Rsrc1, Rsrc2	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0011
I-TYPE																
PUSH Rsrc1	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0	1010
POP	0	0	1	1	1	0	0	0	0	1	0	0	0	1	0	1010

Instruction	RTI	Int	Men_to_Reg	WB	Read_Mem	write_MEM	immediate_value	No_Cond_Branch	branch	SP	Port.flag	Return	Push	Pop	Call	ALU_selection_lines
Rdst	0															
LDM Rdst, Imm	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	1000
LDD Rdst, Rsrc1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	1110
STD	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1000
J-TYPE																
JZ Rdst	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1010
JC Rdst	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1010
JMP Rdst	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1010
CALL Rdst	0	0	0	0	0	0	0	1	0	1	0	0	1	0	1	1110
RET	0	0	0	0	1	0	0	0	0	1	0	1	0	1	0	1010
RTI	1	0	0	0	1	0	0	0	0	1	0	1	0	1	0	1010

ADD	0010
SUB	0001
INC (takes A & B and outputs A+1)	1100
DEC (takes A & B and outputs A-1)	1101
MOV (takes A & B and outputs A)	1110
NOT (takes A & B and outputs not A)	1111

AND	0010
OR	0011
SETC (takes A&B and outputs garbage but changes the carry flag to 1)	0100
CLRC (takes A&B and outputs garbage but changes the carry flag to 0)	0110
LDM (takes A & B and outputs B)	1000
Reset (Make output = 0)	0101
NOP	1010
First time to enter simulation	0000

PC Mux 4x1

0 0	Add 2, From Branching
0 1	Ret, RetI, pc ← dataM[sp]
1 0	PC ← M[0]
1 1	PC ← M[1]

Data1 (Mux 8x1) Output of forward unit

FW[2] FW[1] FW[0]	
0 0 0	Data1
0 1 0	Data from Memory1
1 0 0	Data from Memory 2
1 1 0	Data from WB
0 1 1	Data from in port Memory1
1 0 1	Data from in port Memory2
1 1 1	Data from in port WB

Data2 (Mux 8x1) Output of forward unit

FW[2] FW[1] Fw[0]	
0 0 1	Immediate
0 0 0	Data2
0 1 0	Data from Memory1
1 0 0	Data from Memory2
1 1 0	Data from WB
0 1 1	In port memory 1
1 0 1	In port memory 2
1 1 1	In port WB

Instruction	opCode(6-bits)	1st operand (RSrc1)(3-bits)	2nd operand (Rsrc2)(3-bits)	3rd operand (Rdst)(3-bits)	Extra bit(1-bit)
NOP	000001	xxx	xxx	xxx	x
SETC	000010	xxx	xxx	xxx	x
CLRC	000011	xxx	xxx	xxx	x
OUT Rsrc1	000100	Rsrc1	xxx	OUT.PORT	x
IN Rdst	000101	IN.PORT	xxx	Rdst	x
R-Types					
INC Rdst, Rsrc1	000110	Rsrc1	xxx	Rdst	x
DEC Rdst, Rsrc1	000111	Rsrc1	xxx	Rdst	x
NOT Rdst, Rsrc1	001000	Rsrc1	xxx	Rdst	x
MOV Rdst, Rsrc1	001001	Rsrc1	xxx	Rdst	x
ADD Rdst,	001010	Rsrc1	Rsrc2	Rdst	x

Rsrc1,Rsrc2					
IADD Rdst, Rsrc1, Imm	001011	Rsrc1	xxx	Rdst	x
SUB Rdst, Rsrc1,Rsc2	001100	Rsrc1	Rsc2	Rdst	x
AND Rdst, Rsrc1,Rsrc2	001101	Rsrc1	Rsc2	Rdst	x
OR Rdst, Rsrc1,Rsrc2	001110	Rsrc1	Rsc2	Rdst	x
I-TYPE					
PUSH Rsrc1	001111	Rsrc1	xxx	xxx	x
POP Rdst	010000	xxx	xxx	Rdst	x
LDM Rdst, Imm	010001	xxx	xxx	Rdst	x
LDD Rdst, Rsrc1	010010	Rsrc1	xxx	Rdst	x
STD Rsrc2, Rsrc1	010011	Rsrc1	Rsrc2	xxx	x
J-TYPE					
JZ Rdst	010100	Rdst	xxx	xxx	x
JC Rdst	010101	Rdst	xxx	xxx	x
JMP Rdst	010110	Rdst	xxx	xxx	x
CALL Rdst	010111	Rdst	xxx	xxx	x
RET	011000	xxx	xxx	xxx	x
RTI	011001	xxx	xxx	xxx	x
INT	011010	xxx	xxx	xxx	x

Interrupt sequence:

1) An int signal enters to the instruction memory and the M[1] is out to the FD buffer
2) PC for Int passes through the buffers to the memory stage to be stored in the memory[SP]
3) From the WB stage: the INT signal and M1 enters PC decision unit, and INT enters to be then used to flush the buffers that needs to be flushed
4) At the Execute stage: the CCR is stored