Indian Institute of Technology Bombay
Department of Electrical Engineering

EE224: Digital Design

Prof. Virendra Singh

Autumn Semester 2024

Course Project Report: IITB-CPU

Final Report

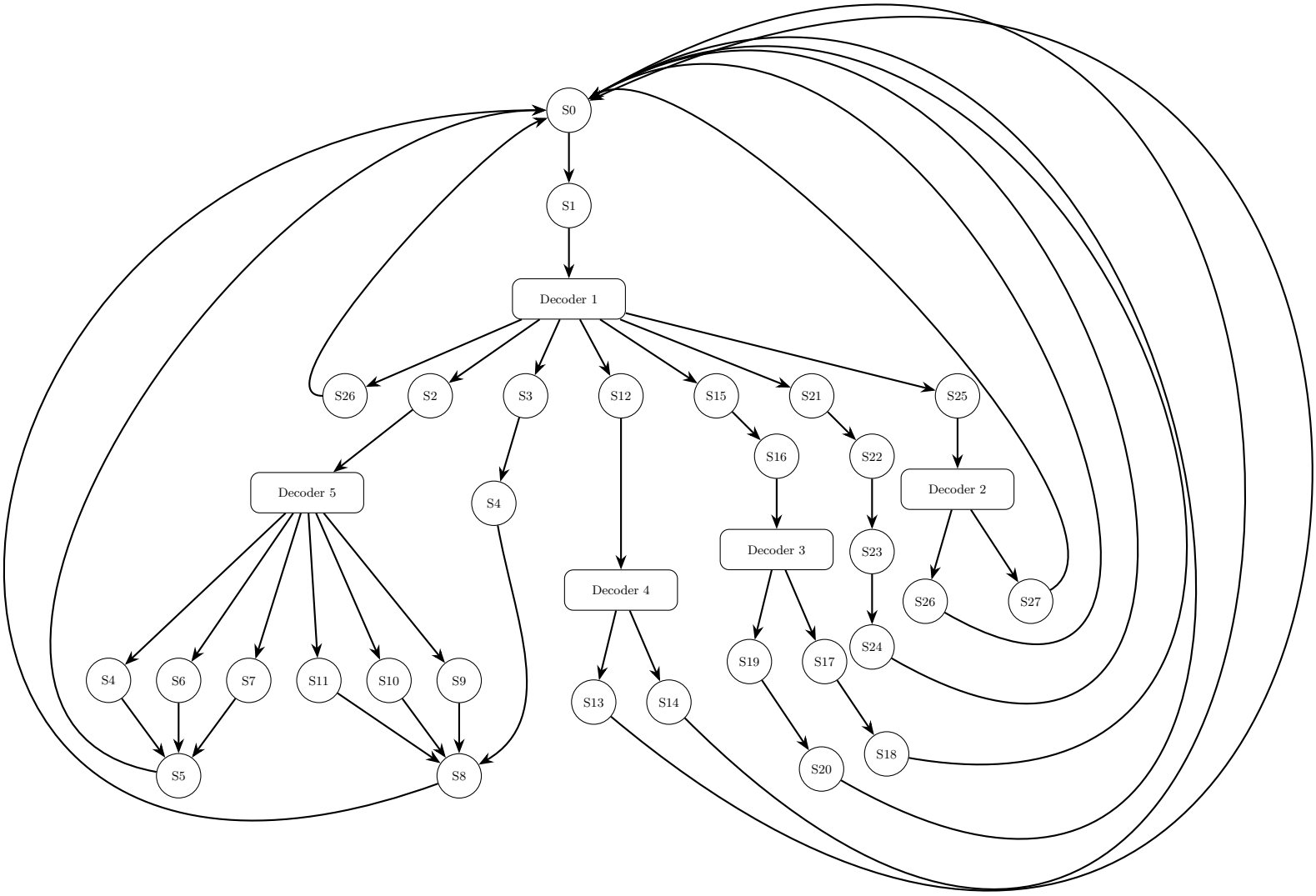
Srivathsa Thotakura | 23B1305
Manepalli Revanth | 23B1223
Nenavath Dheeraj Naik | 23B1248
Sreesti Sahoo | 23B1267
Jadhav Raju | 23B1253

Contents

Chapter 1

Finite State Machine Diagram

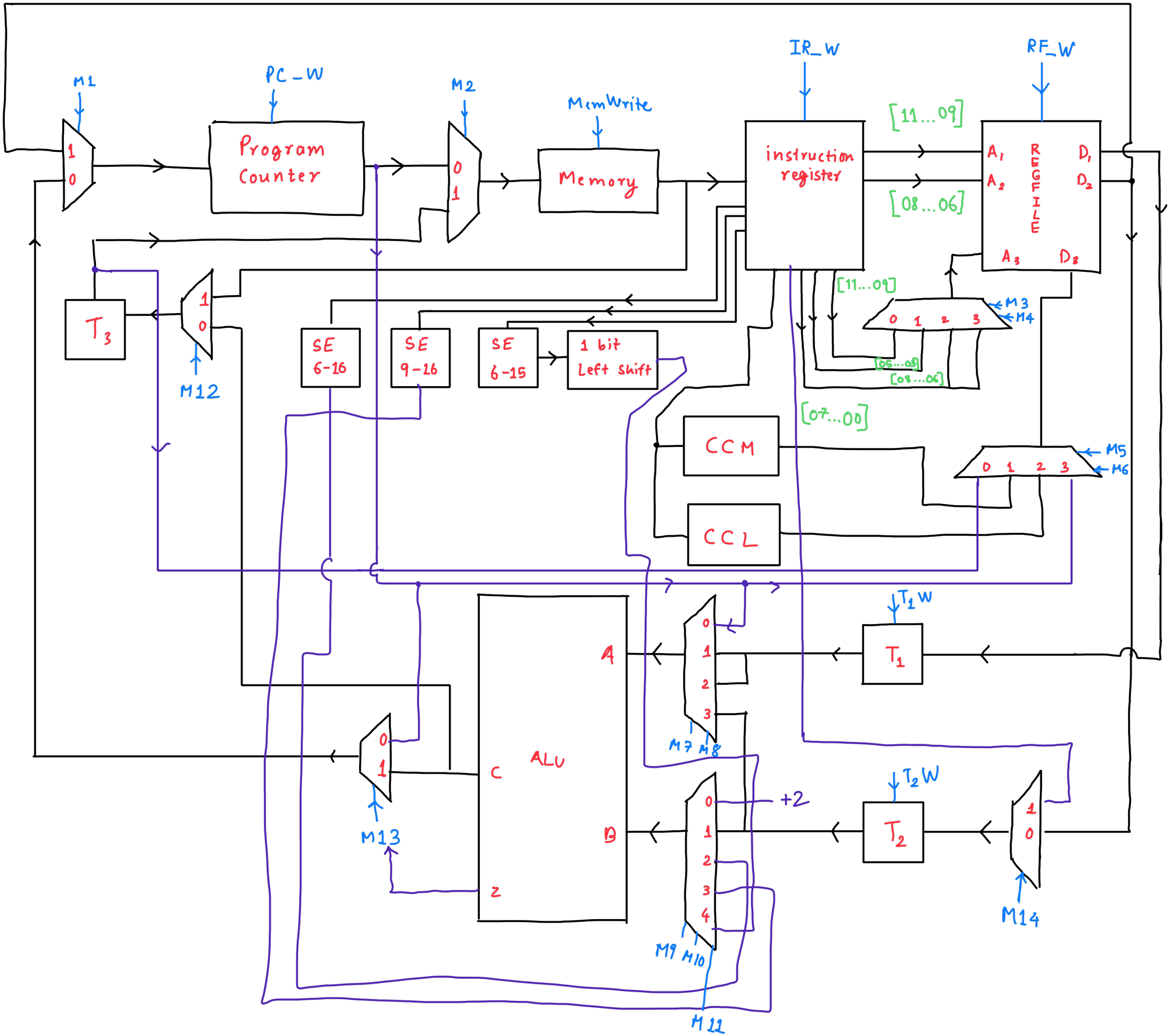
This diagram embodied all the states and the required decoders to integrate all the different datapaths corresponding to the given instructions in the problem statement. States 0-27 are individually defined in the Datapaths chapter.
The below diagram is a Moore machine which can be confirmed simply by the presence of decoders, meaning that the state diagram does not depend on the input but only the state. This state diagram has been realised as digital components in the next chapter.
DIAGRAM ON NEXT PAGE.


S0
S1
Decoder 1
S26
S2
S3
S12
S15
S21
S25
Decoder 5
S4
S16
S22
Decoder 2
Decoder 3
S23
Decoder 4
S26
S27
S4
S6
S7
S11
S10
S9
S19
S17
S24
S13
S14
S18
S5
S8
S20

Chapter 2

IITB CPU Architecture Block Diagram

This architecture has been designed by realising the FSM in the previous chapter. It consists of units like the PC, RF, ALU, etc. SE stands for Sign Extension and CCM & CCL stand for Concatenation at Most Significant Bit and Concatenation at Least Significant Bit, respectively. Multiplexers have been used to implement all the required datapaths. Both black lines and purple lines are the same kind of connections. Essentially they have been kept in a different colour only to avoid confusion. They can be assumed to be analogous to two different layers of PCB routing. DIAGRAM ON NEXT PAGE.

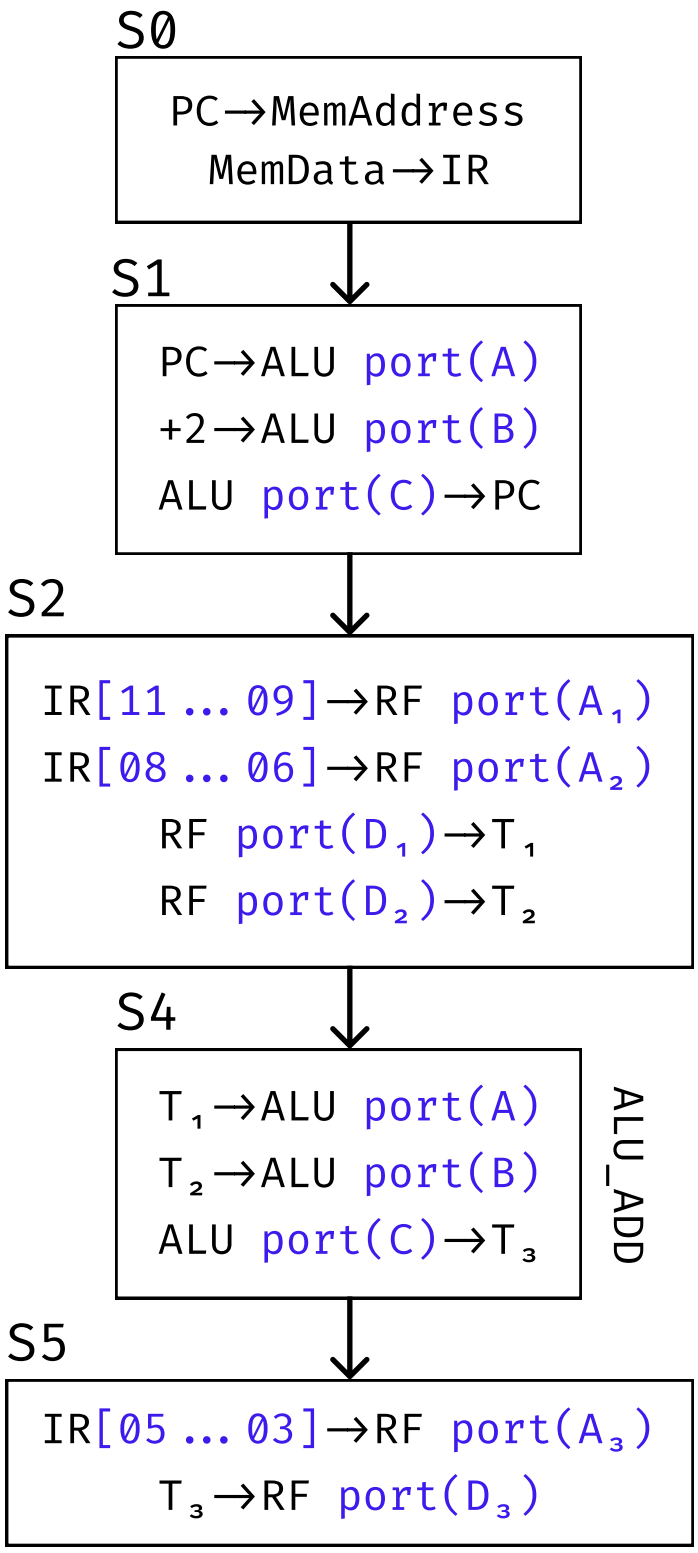

M1
PC_W
M2
MemWrite
IR_W
RF_W
Program Counter
Memory
instruction register
[11...09]
[08...06]
[07...00]
REGFILE
A1
A2
A3
D1
D2
D3
T3
M12
SE 6-16
SE 9-16
SE 6-15
1 bit Left Shift
[11...09]
[05...03]
[08...06]
M3
M4
CCM
CCL
M5
M6
T1W
T1
T2W
T2
A
B
C
Z
ALU
+2
M7
M8
M9
M10
M11
M13
M14

Chapter 3

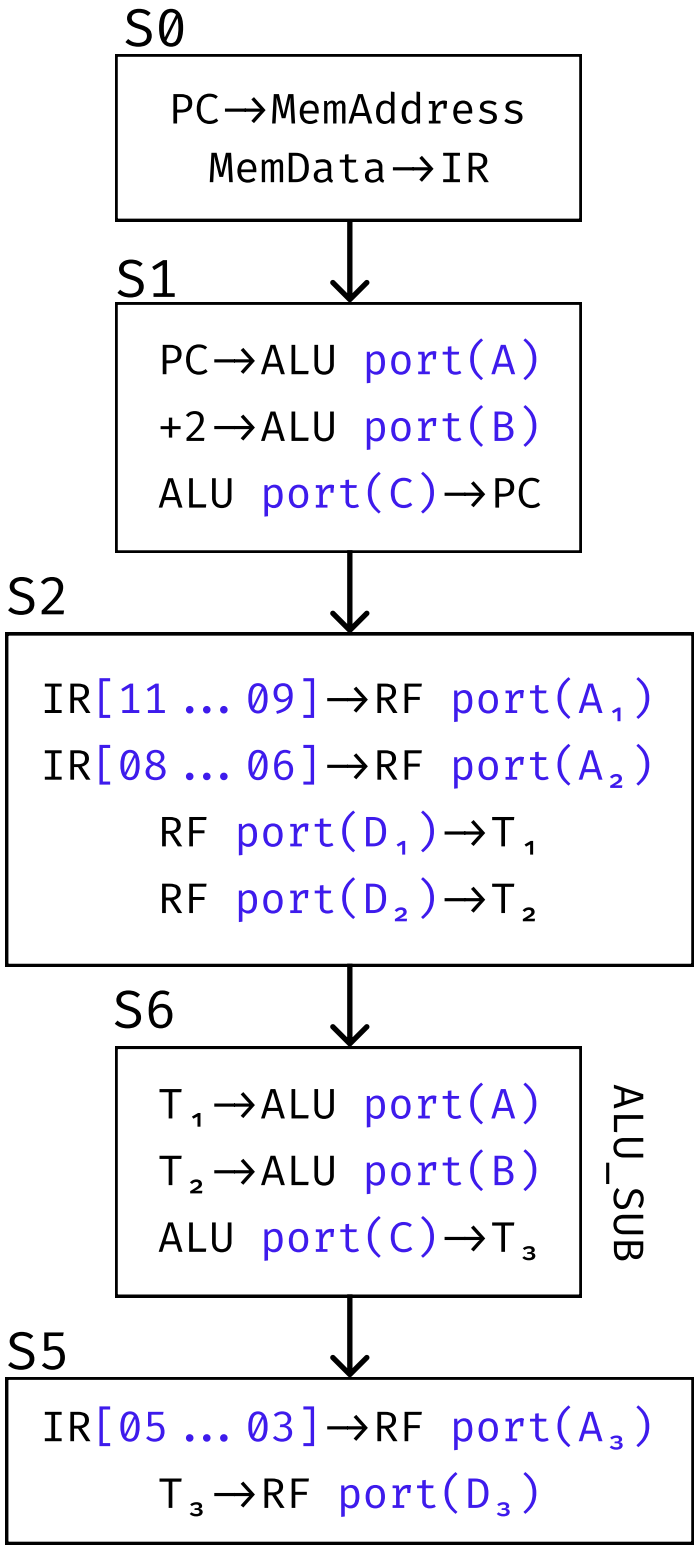
Datapath Schematics

The following are the datapath schematics for the instructions. They will be realised using enable signals that are described in the next chapter. Each instruction's datapath is split into blocks of executions. Common blocks have been merged to create an efficient State Machine. The states have been labeled.

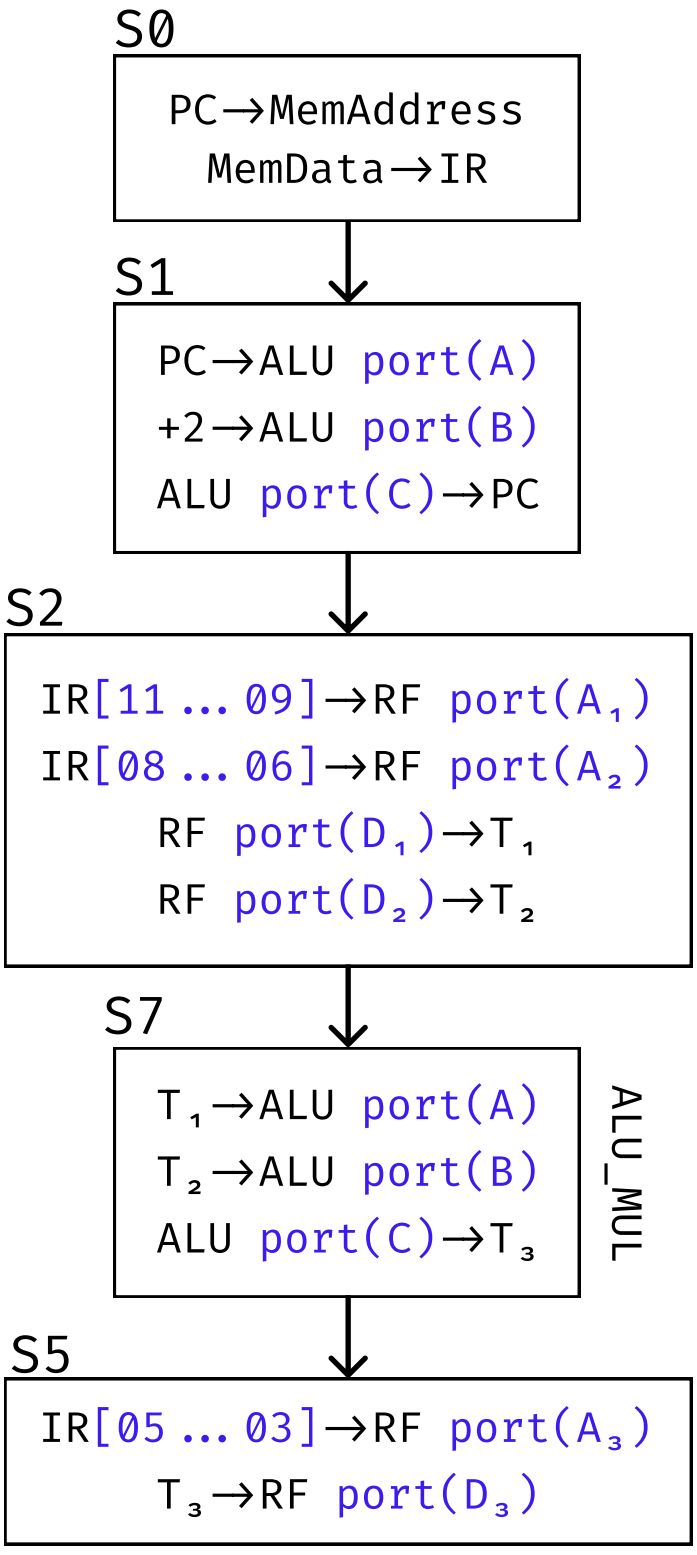
ADD


S0
PC→MemAddress
MemData→IR
S1
PC→ALU port(A)
+2→ALU port(B)
ALU port(C)→PC
S2
IR[11 ... 09]→RF port(A1)
IR[08 ... 06]→RF port(A2)
RF port(D1)→T1
RF port(D2)→T2
S4
T1→ALU port(A)
T2→ALU port(B)
ALU port(C)→T3
ALU_ADD
S5
IR[05 ... 03]→RF port(A3)
T3→RF port(D3)

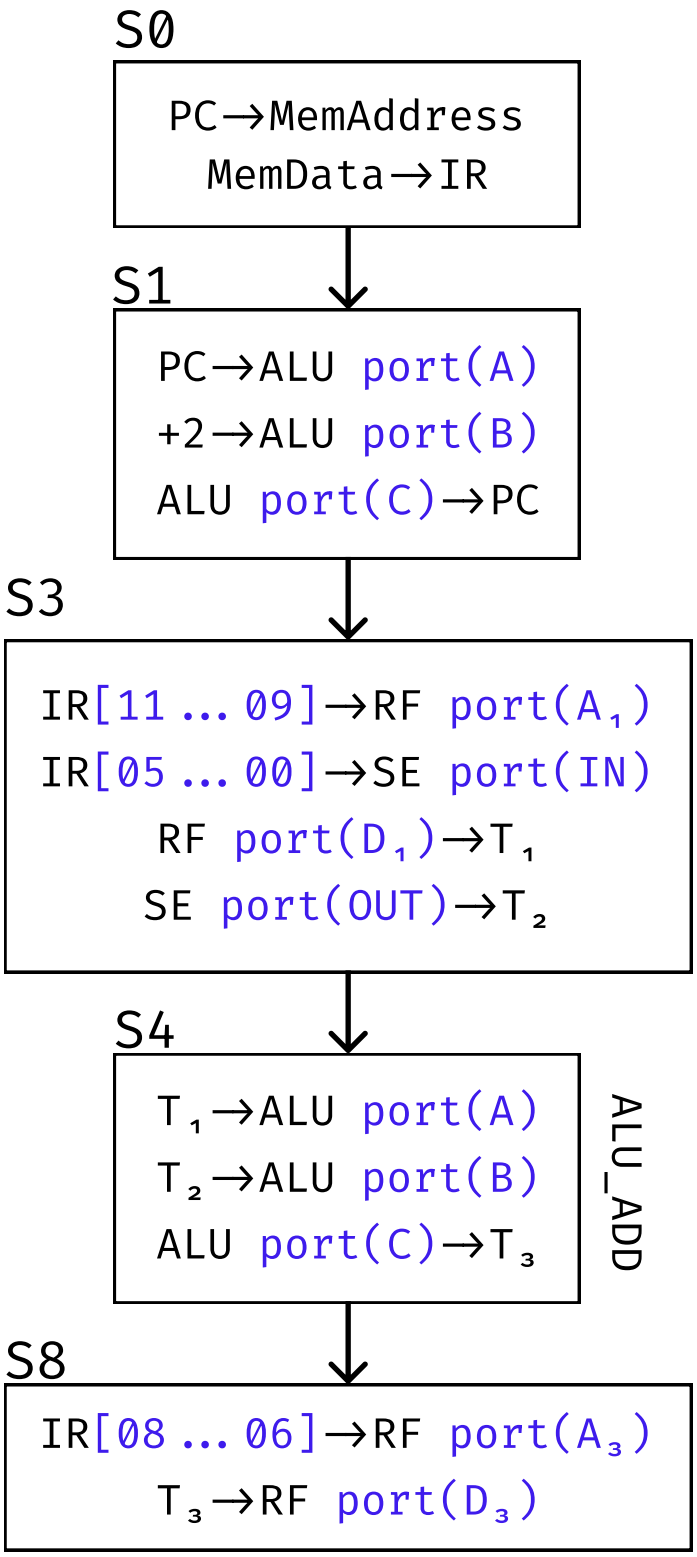
SUB


S0
PC→MemAddress
MemData→IR
S1
PC→ALU port(A)
+2→ALU port(B)
ALU port(C)→PC
S2
IR[11 ... 09]→RF port(A1)
IR[08 ... 06]→RF port(A2)
RF port(D1)→T1
RF port(D2)→T2
S6
T1→ALU port(A)
T2→ALU port(B)
ALU port(C)→T3
ALU_SUB
S5
IR[05 ... 03]→RF port(A3)
T3→RF port(D3)

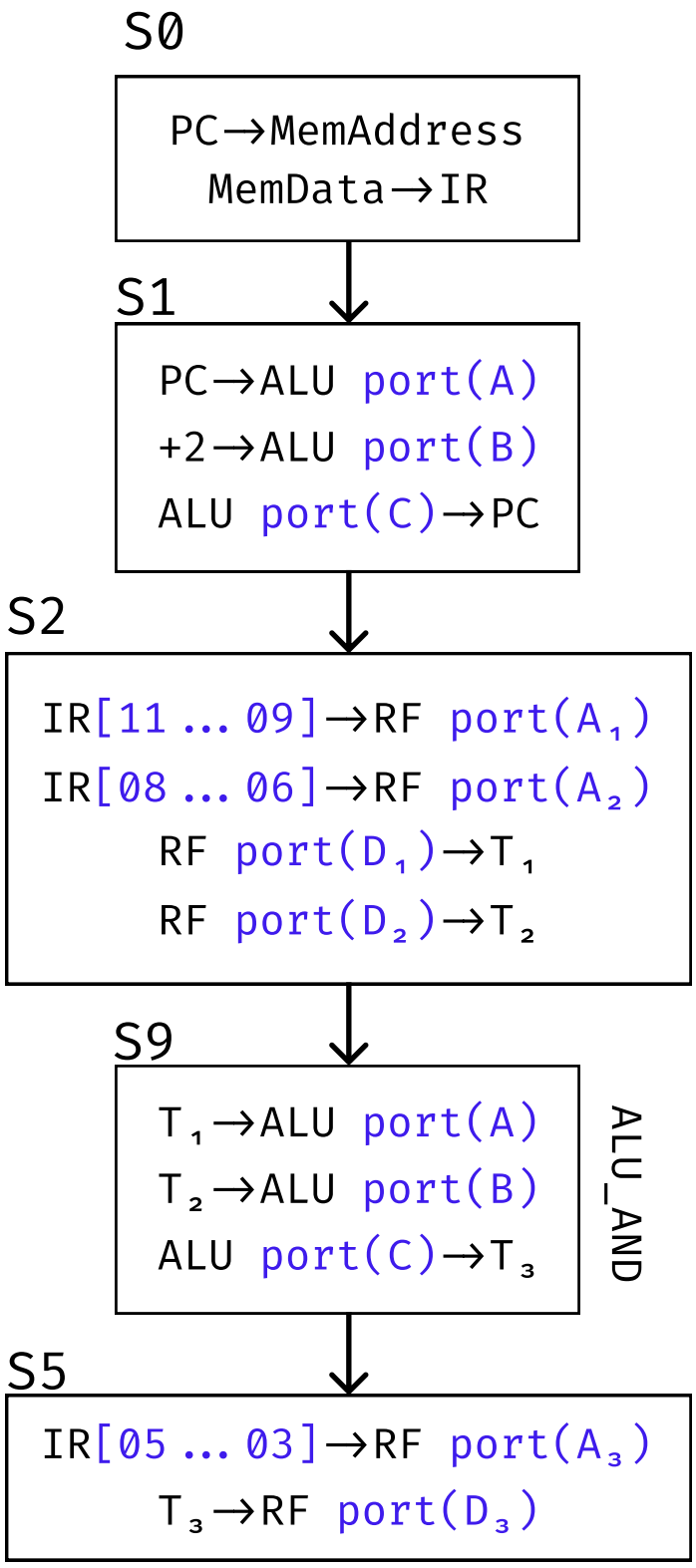
MUL


S0
PC→MemAddress
MemData→IR
S1
PC→ALU port(A)
+2→ALU port(B)
ALU port(C)→PC
S2
IR[11 ... 09]→RF port(A1)
IR[08 ... 06]→RF port(A2)
RF port(D1)→T1
RF port(D2)→T2
S7
T1→ALU port(A)
T2→ALU port(B)
ALU port(C)→T3
ALU_MUL
S5
IR[05 ... 03]→RF port(A3)
T3→RF port(D3)

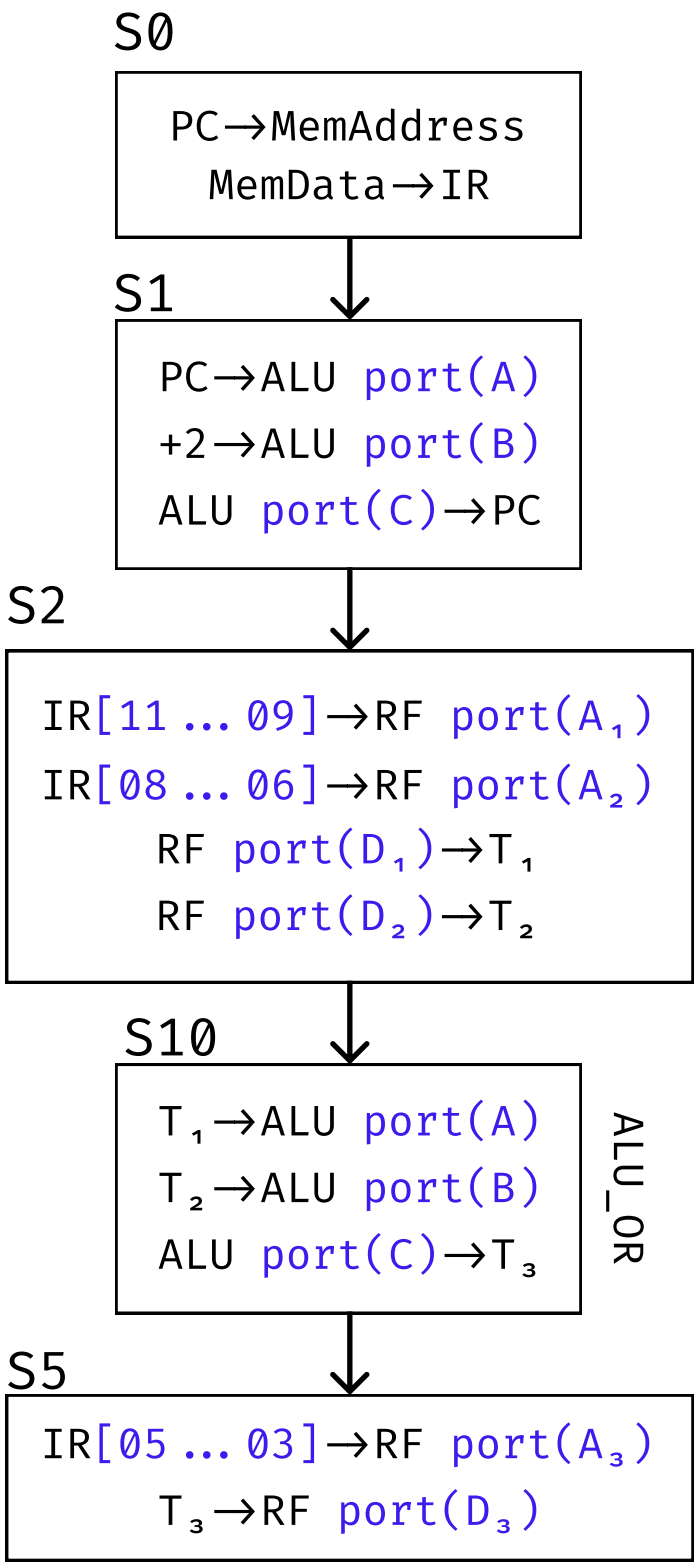
ADI


S0
PC→MemAddress
MemData→IR
S1
PC→ALU port(A)
+2→ALU port(B)
ALU port(C)→PC
S3
IR[11 ... 09]→RF port(A1)
IR[05 ... 00]→SE port(IN)
RF port(D1)→T1
SE port(OUT)→T2
S4
T1→ALU port(A)
T2→ALU port(B)
ALU port(C)→T3
ALU_ADD
S8
IR[08 ... 06]→RF port(A3)
T3→RF port(D3)

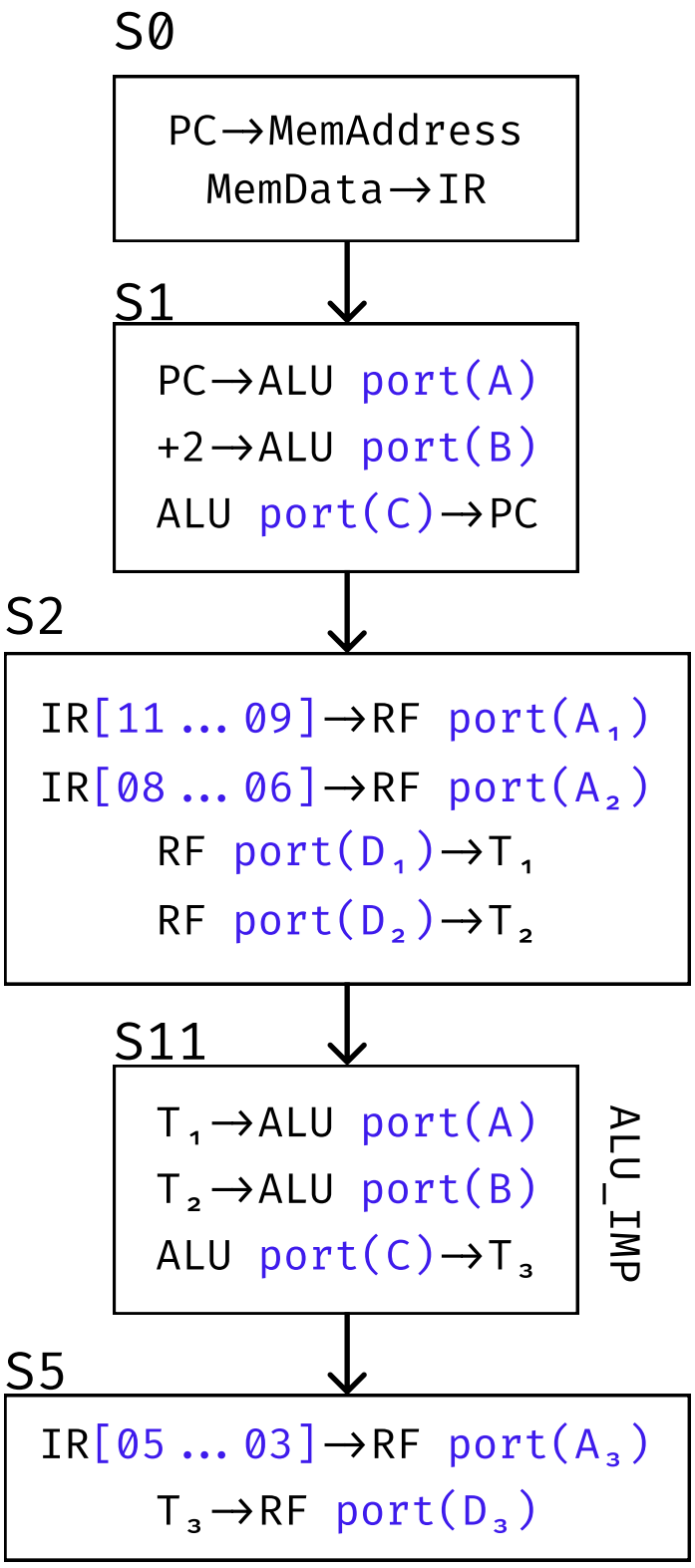
AND


S0
PC→MemAddress
MemData→IR
S1
PC→ALU port(A)
+2→ALU port(B)
ALU port(C)→PC
S2
IR[11 ... 09]→RF port(A1)
IR[08 ... 06]→RF port(A2)
RF port(D1)→T1
RF port(D2)→T2
S9
T1→ALU port(A)
T2→ALU port(B)
ALU port(C)→T3
ALU_AND
S5
IR[05 ... 03]→RF port(A3)
T3→RF port(D3)

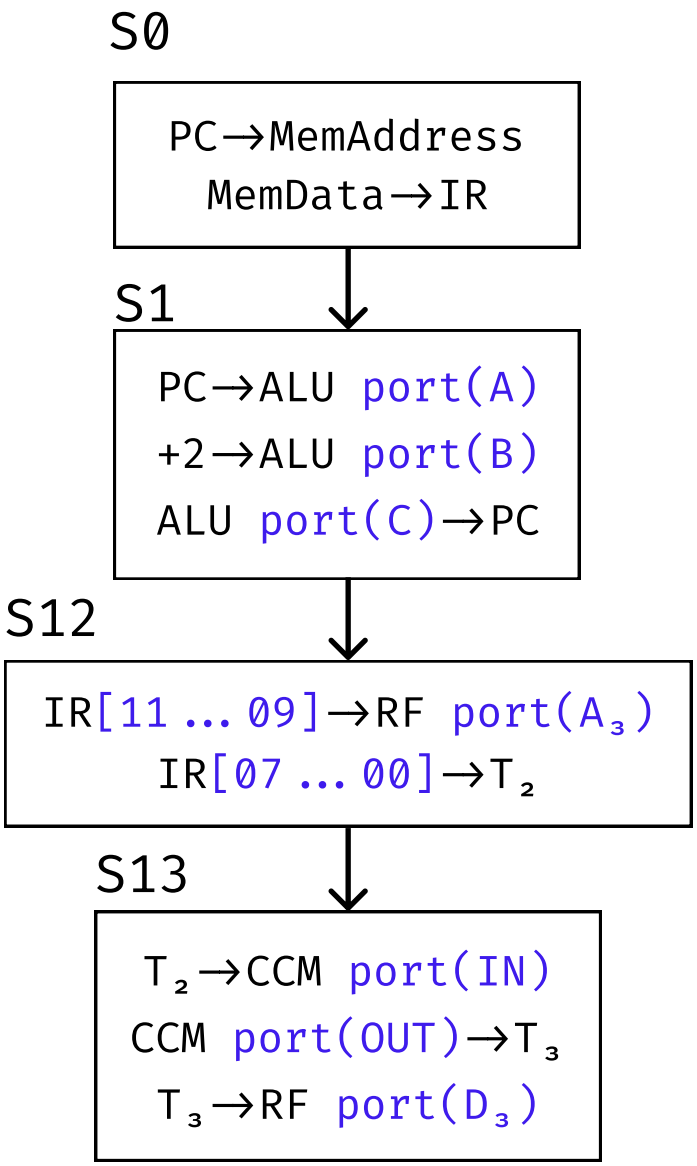
ORA


S0
PC→MemAddress
MemData→IR
S1
PC→ALU port(A)
+2→ALU port(B)
ALU port(C)→PC
S2
IR[11 ... 09]→RF port(A1)
IR[08 ... 06]→RF port(A2)
RF port(D1)→T1
RF port(D2)→T2
S10
T1→ALU port(A)
T2→ALU port(B)
ALU port(C)→T3
ALU_OR
S5
IR[05 ... 03]→RF port(A3)
T3→RF port(D3)

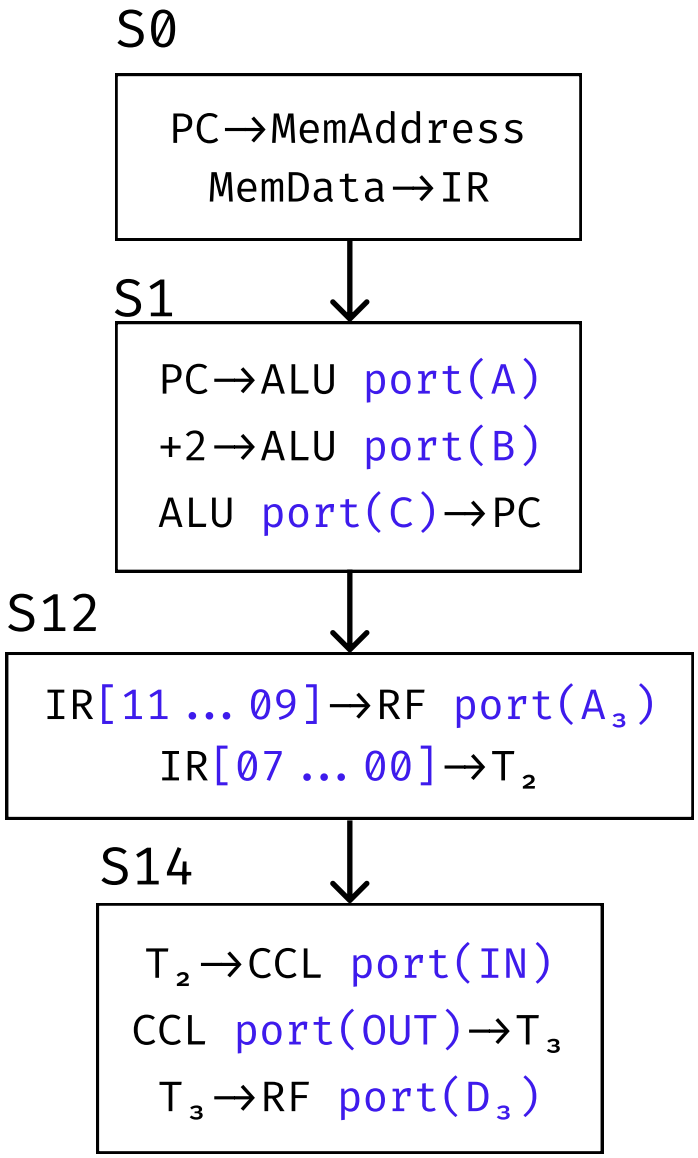
IMP


S0
PC→MemAddress
MemData→IR
S1
PC→ALU port(A)
+2→ALU port(B)
ALU port(C)→PC
S2
IR[11 ... 09]→RF port(A1)
IR[08 ... 06]→RF port(A2)
RF port(D1)→T1
RF port(D2)→T2
S11
T1→ALU port(A)
T2→ALU port(B)
ALU port(C)→T3
ALU_IMP
S5
IR[05 ... 03]→RF port(A3)
T3→RF port(D3)

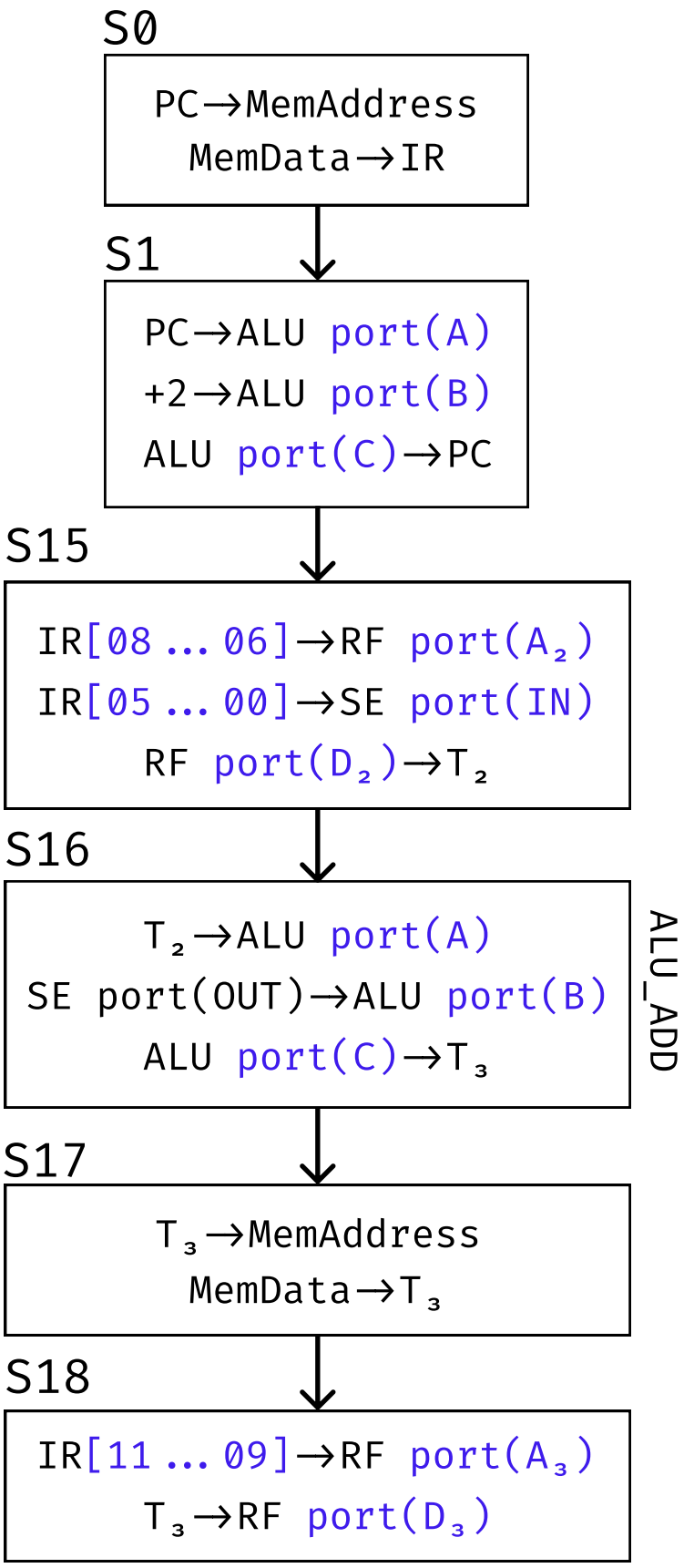
LHI


S0
PC→MemAddress
MemData→IR
S1
PC→ALU port(A)
+2→ALU port(B)
ALU port(C)→PC
S12
IR[11 ... 09]→RF port(A3)
IR[07 ... 00]→T2
S13
T2→CCM port(IN)
CCM port(OUT)→T3
T3→RF port(D3)

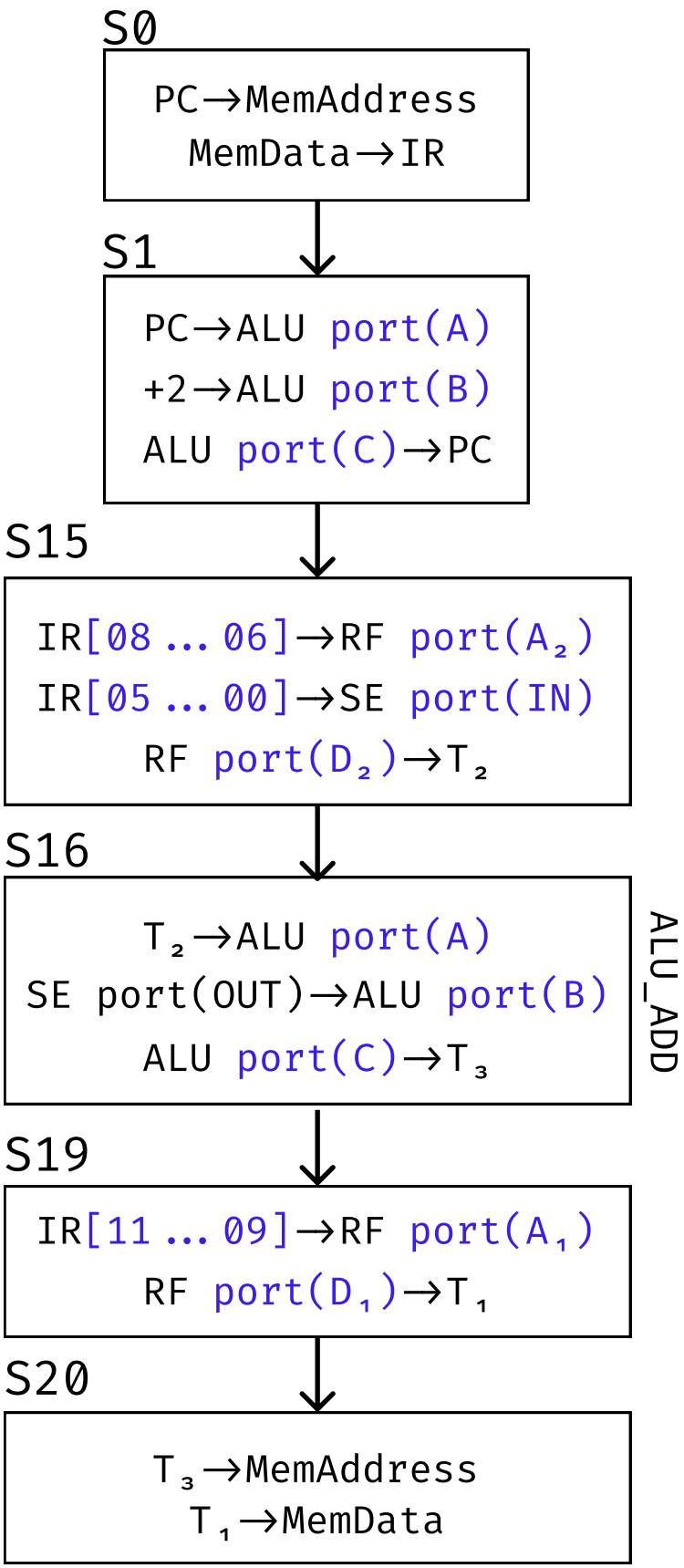
LLI


S0
PC→MemAddress
MemData→IR
S1
PC→ALU port(A)
+2→ALU port(B)
ALU port(C)→PC
S12
IR[11 ... 09]→RF port(A3)
IR[07 ... 00]→T2
S14
T2→CCL port(IN)
CCL port(OUT)→T3
T3→RF port(D3)

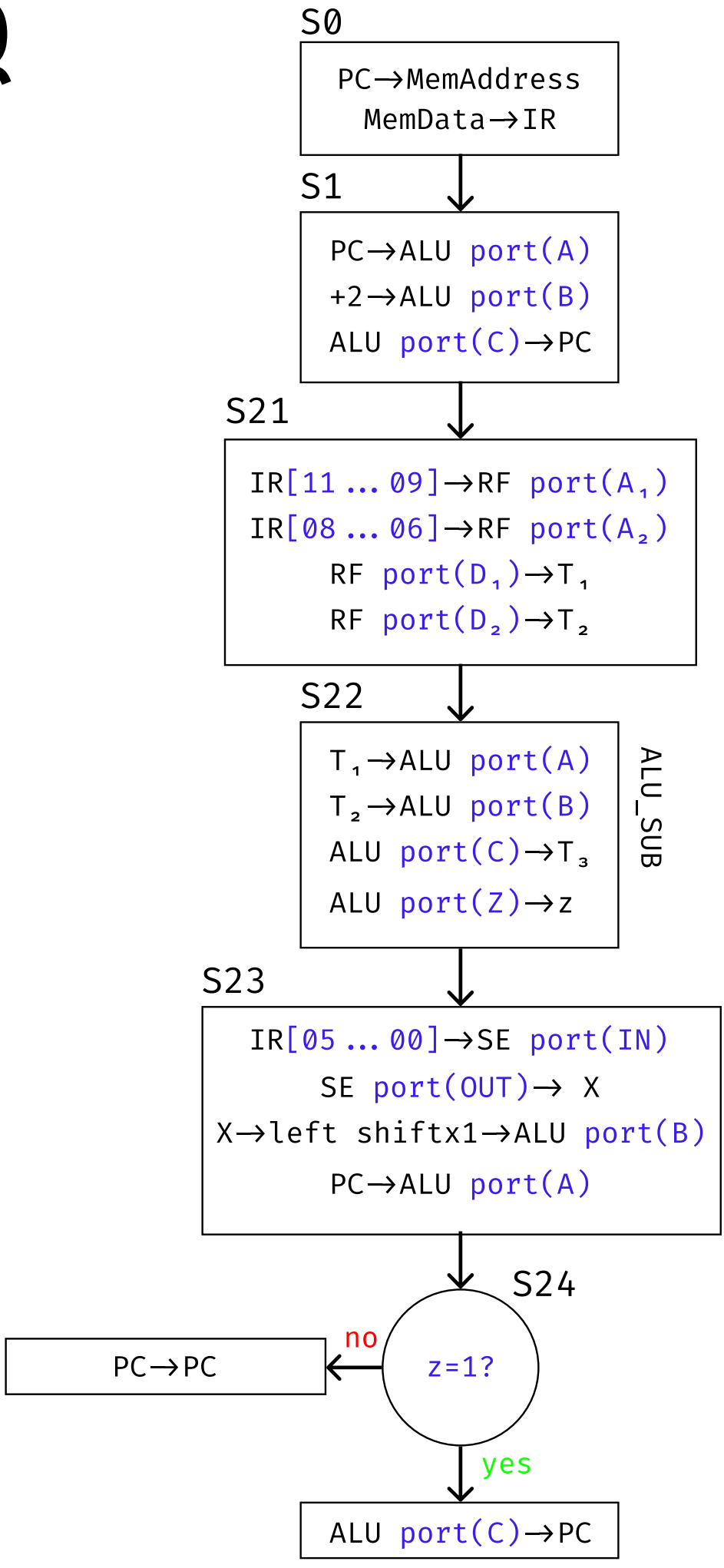
LW


S0
PC→MemAddress
MemData→IR
S1
PC→ALU port(A)
+2→ALU port(B)
ALU port(C)→PC
S15
IR[08 ... 06]→RF port(A2)
IR[05 ... 00]→SE port(IN)
RF port(D2)→T2
S16
T2→ALU port(A)
SE port(OUT)→ALU port(B)
ALU port(C)→T3
ALU_ADD
S17
T3→MemAddress
MemData→T3
S18
IR[11 ... 09]→RF port(A3)
T3→RF port(D3)

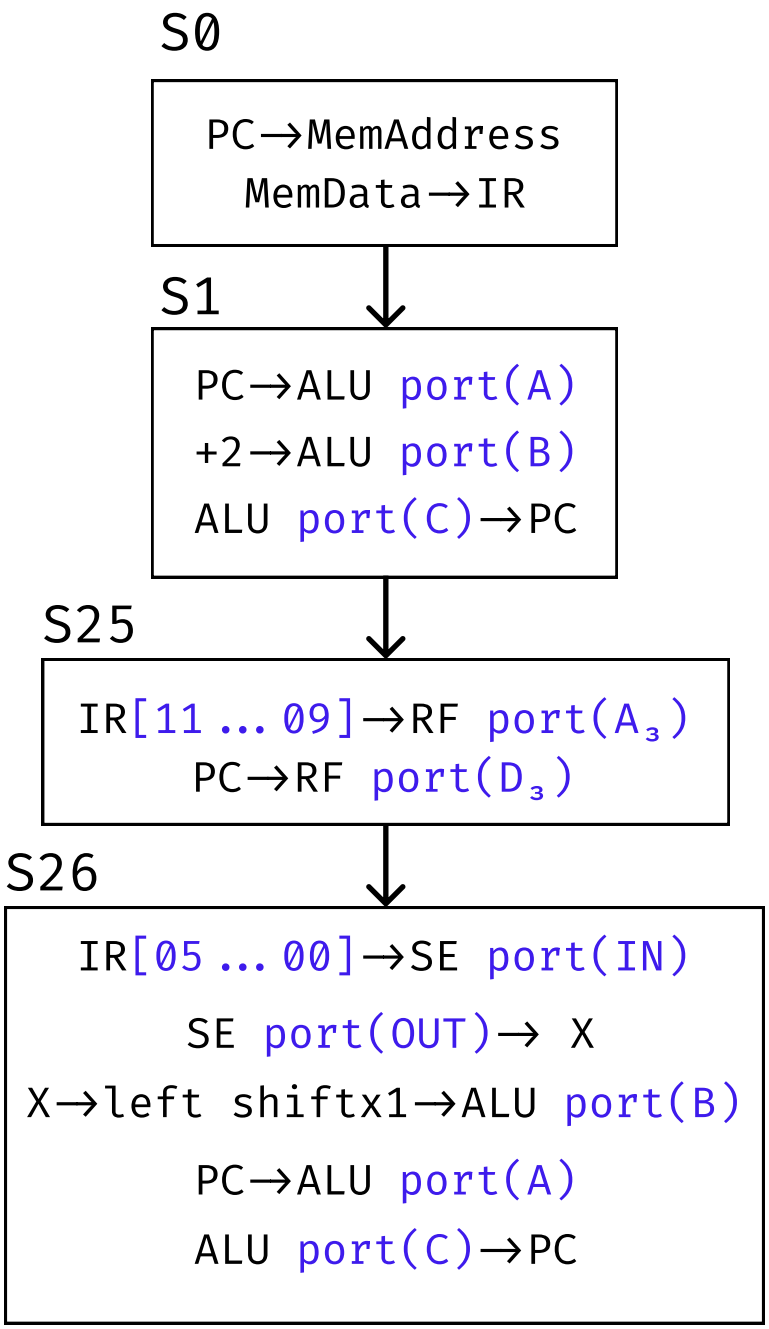
SW


S0
PC→MemAddress
MemData→IR
S1
PC→ALU port(A)
+2→ALU port(B)
ALU port(C)→PC
S15
IR[08 ... 06]→RF port(A2)
IR[05 ... 00]→SE port(IN)
RF port(D2)→T2
S16
T2→ALU port(A)
SE port(OUT)→ALU port(B)
ALU port(C)→T3
ALU_ADD
S19
IR[11 ... 09]→RF port(A1)
RF port(D1)→T1
S20
T3→MemAddress
T1→MemData

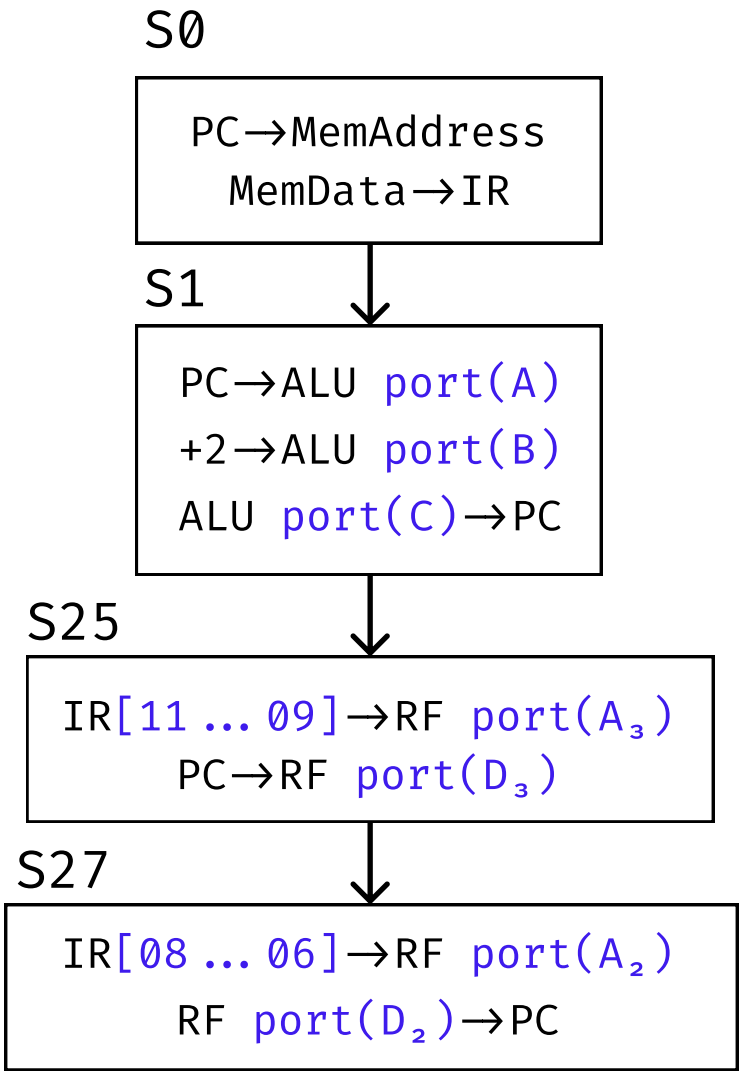
BEQ


S0
PC→MemAddress
MemData→IR
S1
PC→ALU port(A)
+2→ALU port(B)
ALU port(C)→PC
S21
IR[11 ... 09]→RF port(A1)
IR[08 ... 06]→RF port(A2)
RF port(D1)→T1
RF port(D2)→T2
S22
T1→ALU port(A)
T2→ALU port(B)
ALU port(C)→T3
ALU port(Z)→z
ALU_SUB
S23
IR[05 ... 00]→SE port(IN)
SE port(OUT)→ X
X→left shiftx1→ALU port(B)
PC→ALU port(A)
S24
z=1?
no
PC→PC
yes
ALU port(C)→PC

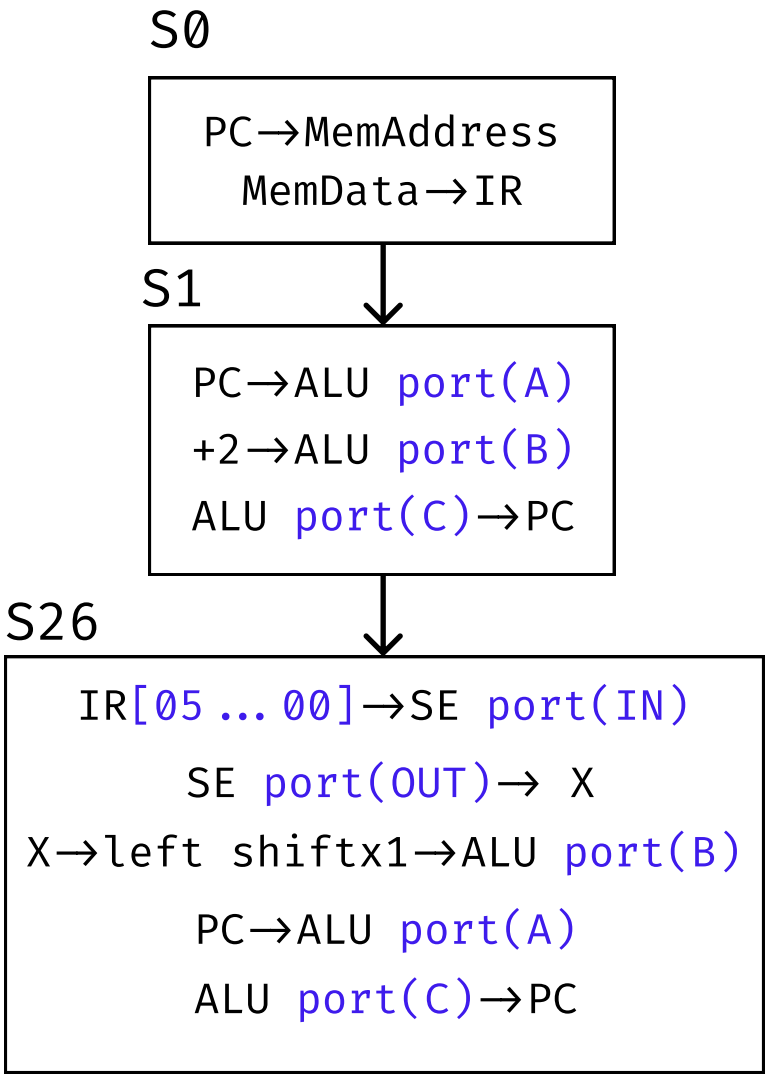
JAL


S0
PC→MemAddress
MemData→IR
S1
PC→ALU port(A)
+2→ALU port(B)
ALU port(C)→PC
S25
IR[11 ... 09]→RF port(A3)
PC→RF port(D3)
S26
IR[05 ... 00]→SE port(IN)
SE port(OUT)→ X
X→left shiftx1→ALU port(B)
PC→ALU port(A)
ALU port(C)→PC

JLR


S0
PC→MemAddress
MemData→IR
S1
PC→ALU port(A)
+2→ALU port(B)
ALU port(C)→PC
S25
IR[11 ... 09]→RF port(A3)
PC→RF port(D3)
S27
IR[08 ... 06]→RF port(A2)
RF port(D2)→PC

J


S0
PC→MemAddress
MemData→IR
S1
PC→ALU port(A)
+2→ALU port(B)
ALU port(C)→PC
S26
IR[05 ... 00]→SE port(IN)
SE port(OUT)→ X
X→left shiftx1→ALU port(B)
PC→ALU port(A)
ALU port(C)→PC

Chapter 4

State and Control Signal Relationship Table

The overall architecture is being implemented using multiplexers and elements that use enable signals to work. Hence, at the top level we need to define our working using these signals and the Finite States. On the next page is a table that deals with appropriate signal's activity depending on the state of the system.

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	PC-W	MW	IR-W	RF-W	T1-W	T2-W	T3-W	Z-EN
S0																	HIGH					
S1													HIGH		HIGH							
S2																			HIGH	HIGH		
S3																			HIGH			
S4								HIGH			HIGH										HIGH	
S5				HIGH														HIGH				
S6								HIGH			HIGH										HIGH	
S7								HIGH			HIGH										HIGH	
S8			HIGH															HIGH				
S9								HIGH			HIGH										HIGH	
S10								HIGH			HIGH										HIGH	
S11								HIGH			HIGH		HIGH								HIGH	
S12														HIGH						HIGH		
S13																		HIGH			HIGH	
S14																		HIGH			HIGH	
S15																				HIGH		
S16							HIGH	HIGH		HIGH											HIGH	
S17		HIGH										HIGH										

S18																		HIGH				
S19																			HIGH			
S20		HIGH														HIGH						
S21																			HIGH	HIGH		
S22								HIGH			HIGH											
S23									HIGH													
S24														Z								HIGH
S25					HIGH	HIGH												HIGH				
S26									HIGH				HIGH									
S27	HIGH																					

Chapter 5

Test Case Outputs

Bibliography

We have taken help from our class notes of EE224 and the book "Computer Organisation and Design" by David A. Patterson and John L. Hennessy.