

FAST (Fast Assembly Super Turbo)

ISA

Part A. ISA Introduction

FAST (stands for Fast Assembly Super Turbo) ISA

Philosophy is to minimize use of loops except for looping through entries and have some instructions carry implicit details. There are some very specific instructions such as score and the add, sub. There is also use of unique machine codes mapped to instruction types to maximize versatility with limited number of bits.

1. Instruction list

Instruction	PC	Coding	Functionality	Example	
init Rx, imm	PC++	000 x iii	Rx = imm Rx ∈ {R0, R1} imm: [1,8]	init R0,111	000 0 111
ld Rx, Ry	PC++	001 xx yy	Rx = Mem[Ry]	ld R0, R1	001 00 01
str Rx, Ry	PC++	010 xx yy	Mem[Ry] = Rx	st R0, R1	010 00 01
addR Rx	PC++	01100 xx	R2 = Rx + Rx	addR R0	01100 00
addR2 Rx	PC++	01110 xx	R2 = R2 + Rx	addR2 R0	01110 00
addR3 Rx	PC++	01111 xx	R3 = Rx + Rx	addR3 R0	01111 00
subR3 Rx	PC++	01101 xx	R3 = R3 - Rx	subR3 R0	01101 00
addi Rx imm	PC++	100 xx ii	Rx = Rx + imm	addi R0, 2	100 00 10
sltR0 Rx,Ry	PC++	101 xx yy	R0 = 1 if Rx < Ry	sltR0 R0,R1	101 00 11
beqR0 Rx imm	if Rx==R0: PC == MUX(imm) else: PC++	11 xx iii	Rx ∈ {R0, R1, R3} Imm number will go into a MUX to select specific jumps	beqR0 R0,3	11 00 011

scrR3R2	PC++	1110 111	R3 = the match score of R3 and R2. This function is done using logic circuit.	scrR3R2	1110 111
---------	------	----------	---	---------	----------

2. Register Design

Register Name	Number
R0	00
R1	01
R2	10
R3	11

3. Control Flow

Since there are just several branches used in our Program1 and Program2, the instruction address of all these branches are constant, we save all these address into a MUX, and use the immediate number in the beq instructions to select it.

Accordingly, there is no need for us to calculate the target addresses.

4. Memory Model

4.1 Data Memory

- 16-bit double-byte addressable
- 128 memory units in total
- using 7-bit address.

Address	Memory
000 0000	Mem[0]
000 0001	Mem[1]
...	...
111 1111	Mem[127]

4.2 Instruction Memory

- 8-bit byte addressable, PC is initialized at 0
- 64 memory units in total
- using 6-bit address.

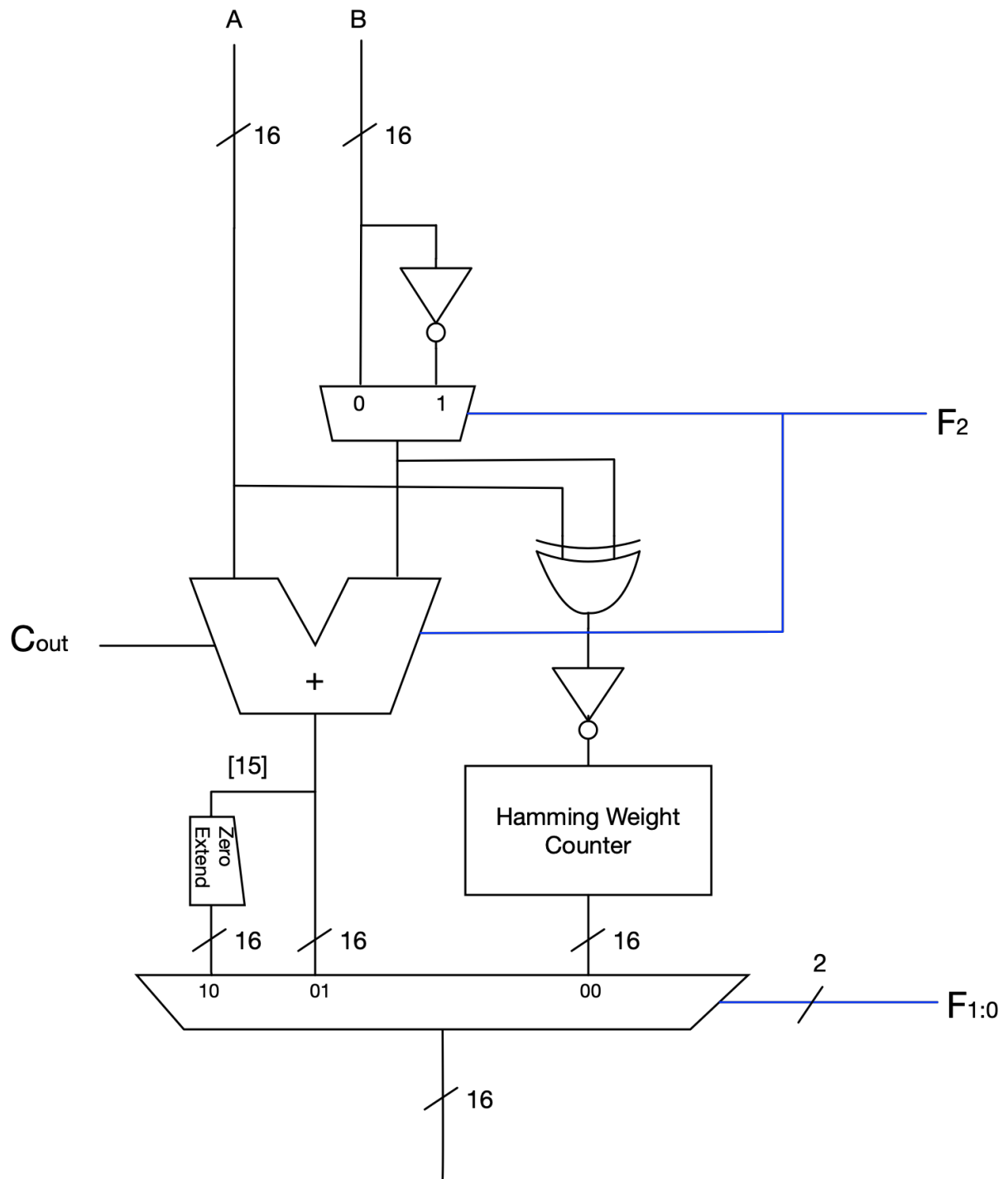
Address	Memory
00 0000	Mem[0]
00 0001	Mem[1]
...	...
11 1111	Mem[63]

Part B. Answers to Questions

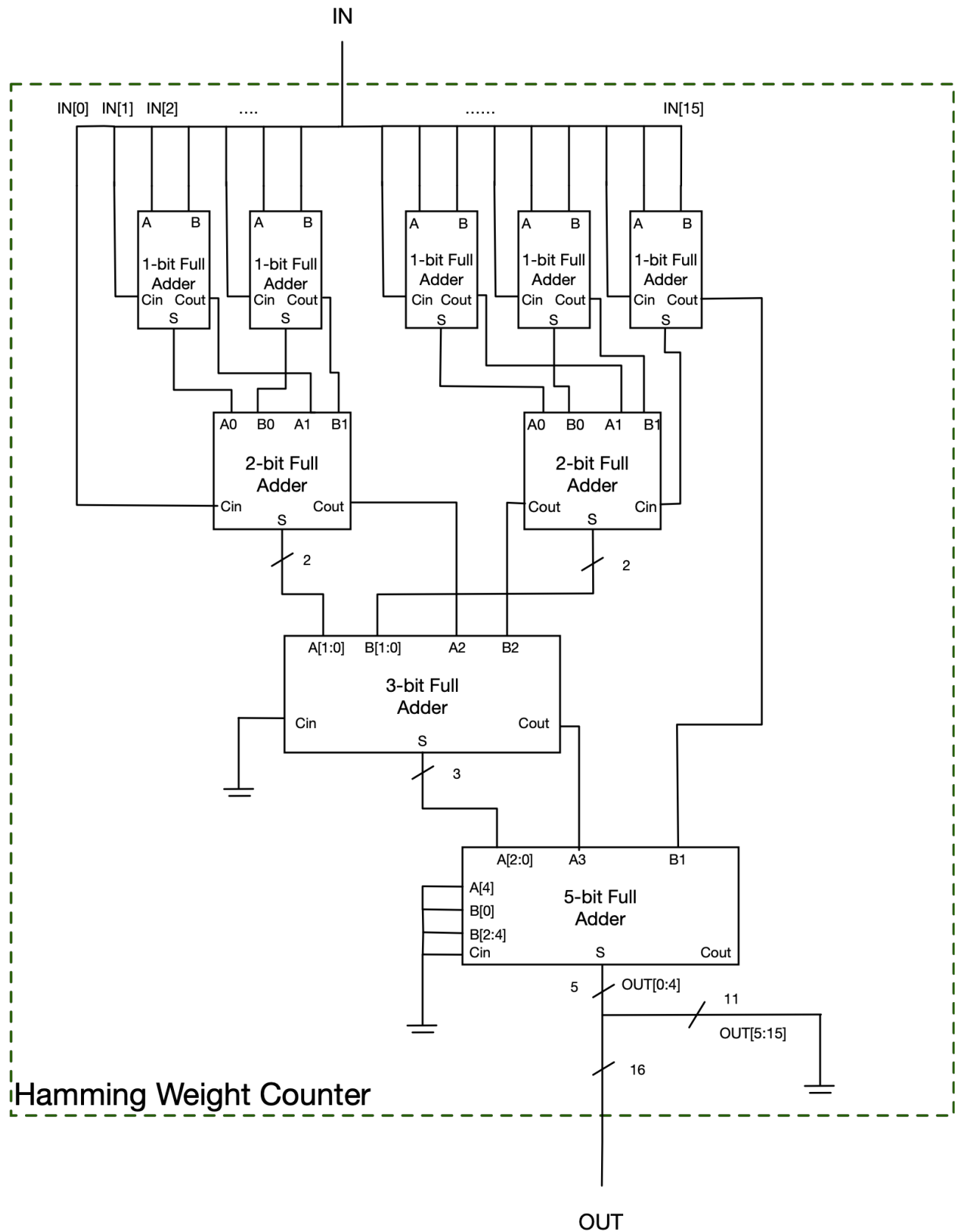
Part C. Software Package

Part D. Hardware Implementation

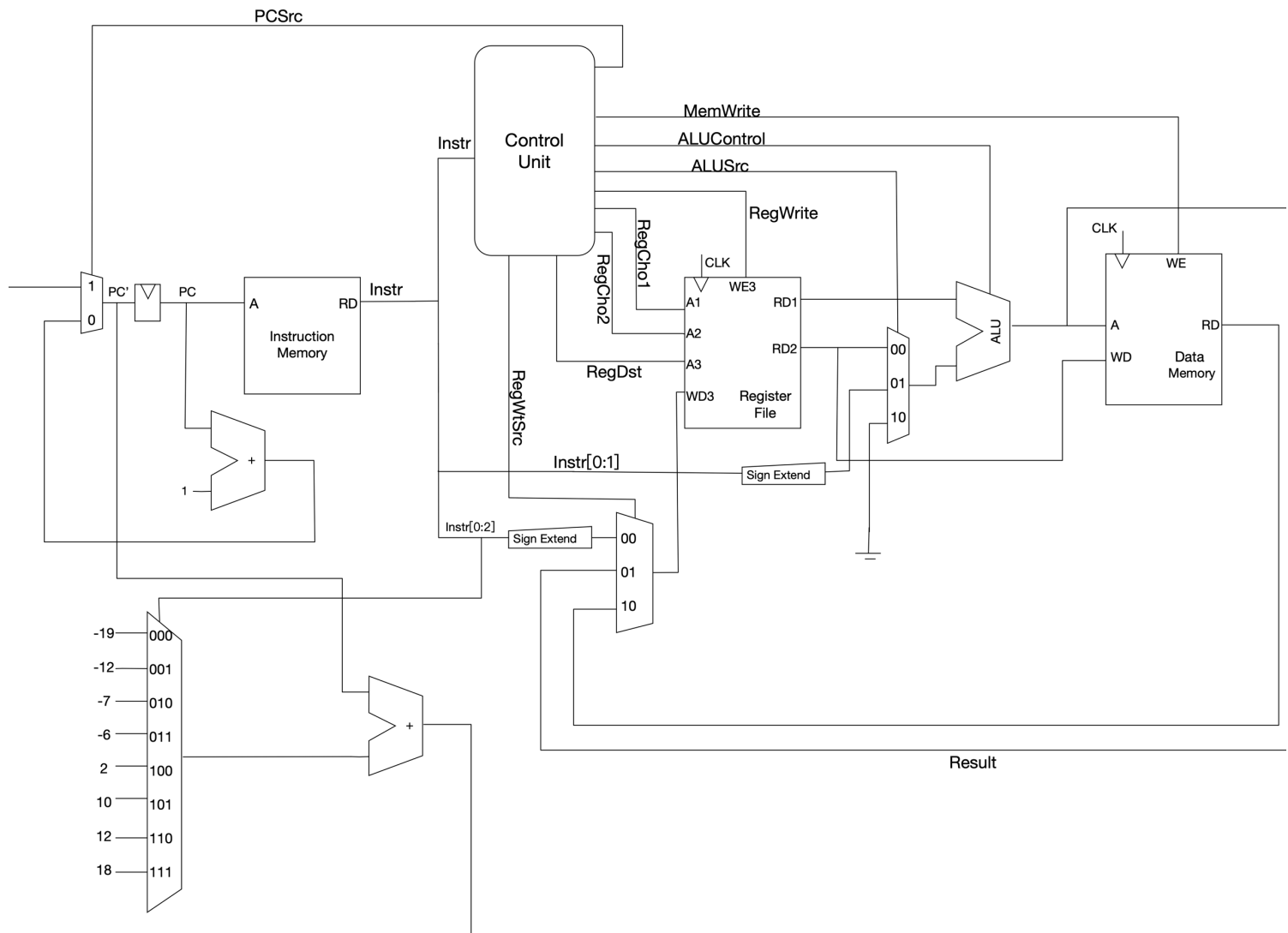
1. ALU schematic



Hamming Weight Counter



2. CPU Datapath



3. Control Logic

Instr	Op	PCSrc	MemWrite	ALUControl	ALUSrc	RegWrite	RegCho1	RegCho2	RegDst	RegWtSrc
init	000 r iii	0	0	XXX	XX	1	XX	XX	0r	00
ld	001 rr ss	0	0	001	10	1	ss	XX	rr	10
str	010 rr ss	0	1	001	10	0	ss	rr	XX	XX
addR	01100 rr	0	0	001	00	1	rr	rr	10	01
addR2	01110 rr	0	0	001	00	1	10	rr	10	01
addR3	01111 rr	0	0	001	00	1	rr	rr	11	01
subR3	01101 rr	0	0	101	00	1	11	rr	11	01
addi	100 rr ii	0	0	001	01	1	rr	XX	rr	01
sltR0	101 rr ss	0	0	110	00	1	rr	ss	00	01
beqR0	11 rr iii	1	0	101	00	0	rr	00	XX	XX
scrR3R2	1110 111	0	0	000	00	1	10	11	11	01