Computer Architecture Take Home Final Exam

The following final is to be completed in private. You are not to discuss any aspects of this exam with any students prior to 5/5/18 at 09:00. You are permitted to use your laptop, development boards, development tools, books, the internet, etc. You are forbidden from asking (verbal, text, phone, IM, forums, etc.) for help (with regard to this exam) from any resource. Violation of these rules will result in a 0 grade.

Build a Simple Single-Cycle Computer (Figure 1) which supports the instruction set in Figure 2 with the instruction format shown in Figure 3. Given these requirements a control unit decoder has already been designed (Figure 4)

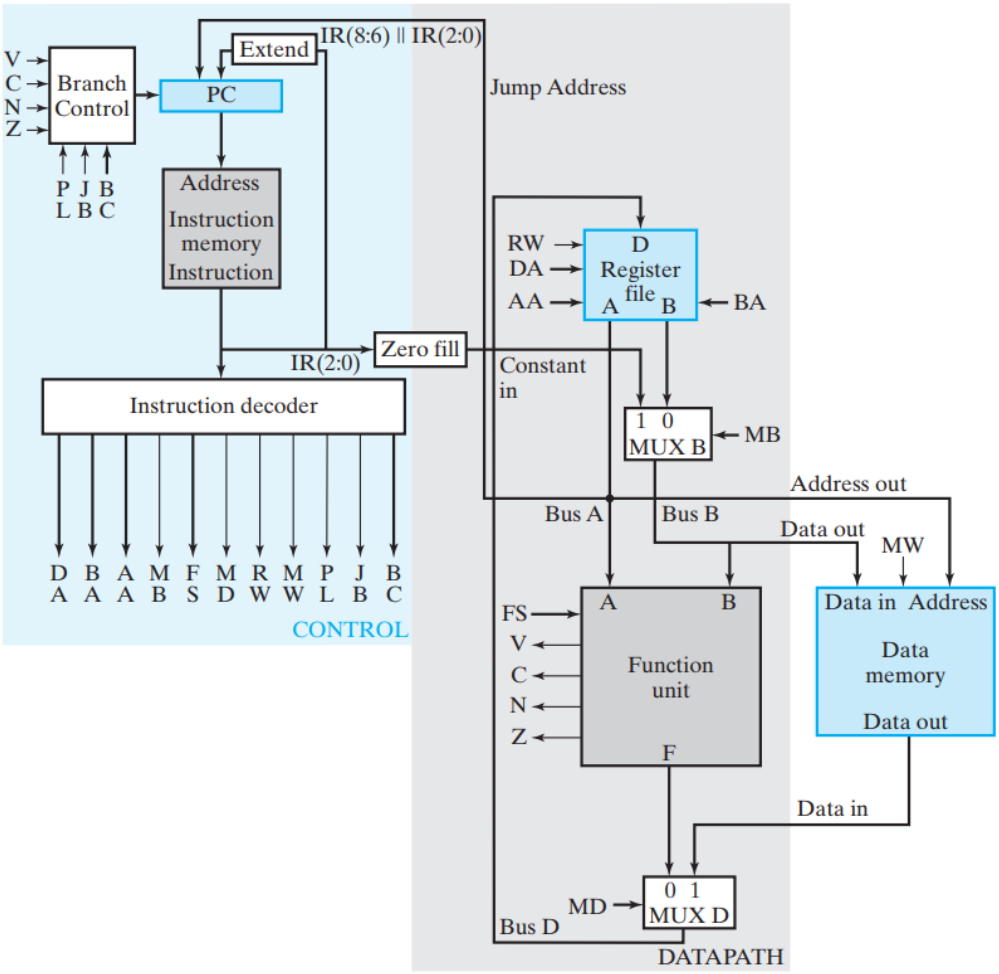

Figure 1 *– Simple Single Cycle CPU*

Instruction Specifications for the Simple Computer

Instruction	Opcode	Mnemonic	Format	Description	Status Bits
Move A	0000000	MOVA	RD, RA	R[DR] ← R[SA]*	N, Z
Increment	0000001	INC	RD, RA	R[DR] ← R[SA] + 1*	N, Z
Add	0000010	ADD	RD, RA, RB	R[DR] ← R[SA] + R[SB]*	N, Z
Subtract	0000101	SUB	RD, RA, RB	R[DR] ← R[SA] – R[SB]*	N, Z
Decrement	0000110	DEC	RD, RA	R[DR] ← R[SA] – 1*	N, Z
AND	0001000	AND	RD, RA, RB	R[DR] ← R[SA] ∧ R[SB]*	N, Z
OR	0001001	OR	RD, RA, RB	R[DR] ← R[SA] ∨ R[SB]*	N, Z
Exclusive OR	0001010	XOR	RD, RA, RB	R[DR] ← R[SA] ⊕ R[SB]*	N, Z
NOT	0001011	NOT	RD, RA	R[DR] ← $\overline{\text{R[SA]}}$ *	N, Z
Move B	0001100	MOVB	RD, RB	R[DR] ← R[SB]*	
Shift Right	0001101	SHR	RD, RB	R[DR] ← sr R[SB]*	
Shift Left	0001110	SHL	RD, RB	R[DR] ← sl R[SB]*	
Load Immediate	1001100	LDI	RD, OP	R[DR] ← zf OP*	
Add Immediate	1000010	ADI	RD, RA, OP	R[DR] ← R[SA] + zf OP*	N, Z
Load	0010000	LD	RD, RA	R[DR] ← M[SA]*	
Store	0100000	ST	RA, RB	M[SA] ← R[SB]*	
Branch on Zero	1100000	BRZ	RA, AD	if (R[SA] = 0) PC ← PC + se AD, if (R[SA] ≠ 0) PC ← PC + 1	N, Z
Branch on Negative	1100001	BRN	RA, AD	if (R[SA] < 0) PC ← PC + se AD, if (R[SA] ≥ 0) PC ← PC + 1	N, Z
Jump	1110000	JMP	RA	PC ← R[SA]	

* For all of these instructions, PC ← PC + 1 is also executed to prepare for the next cycle.

Figure 2 *– Table of instructions for the Simple Single Cycle CPU*

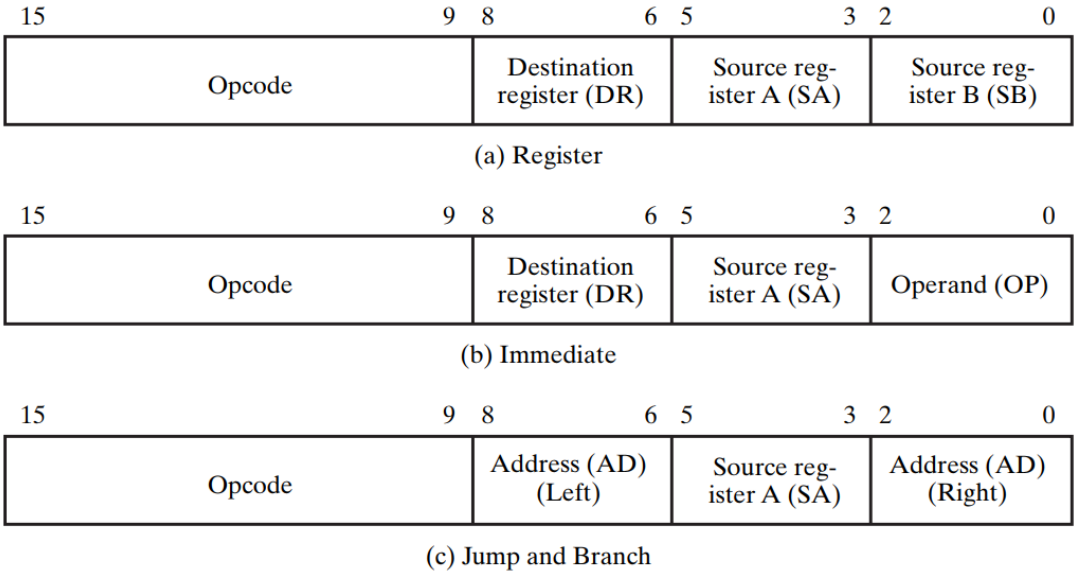

Figure 3 *– Instruction formats for the Simple Single Cycle CPU – Formats with RD, RA and/or RB should use (a) Register; Formats which include OP should use (b) Immediate; Formats for BRZ, BRN & JMP should use (c) Jump and Branch.*

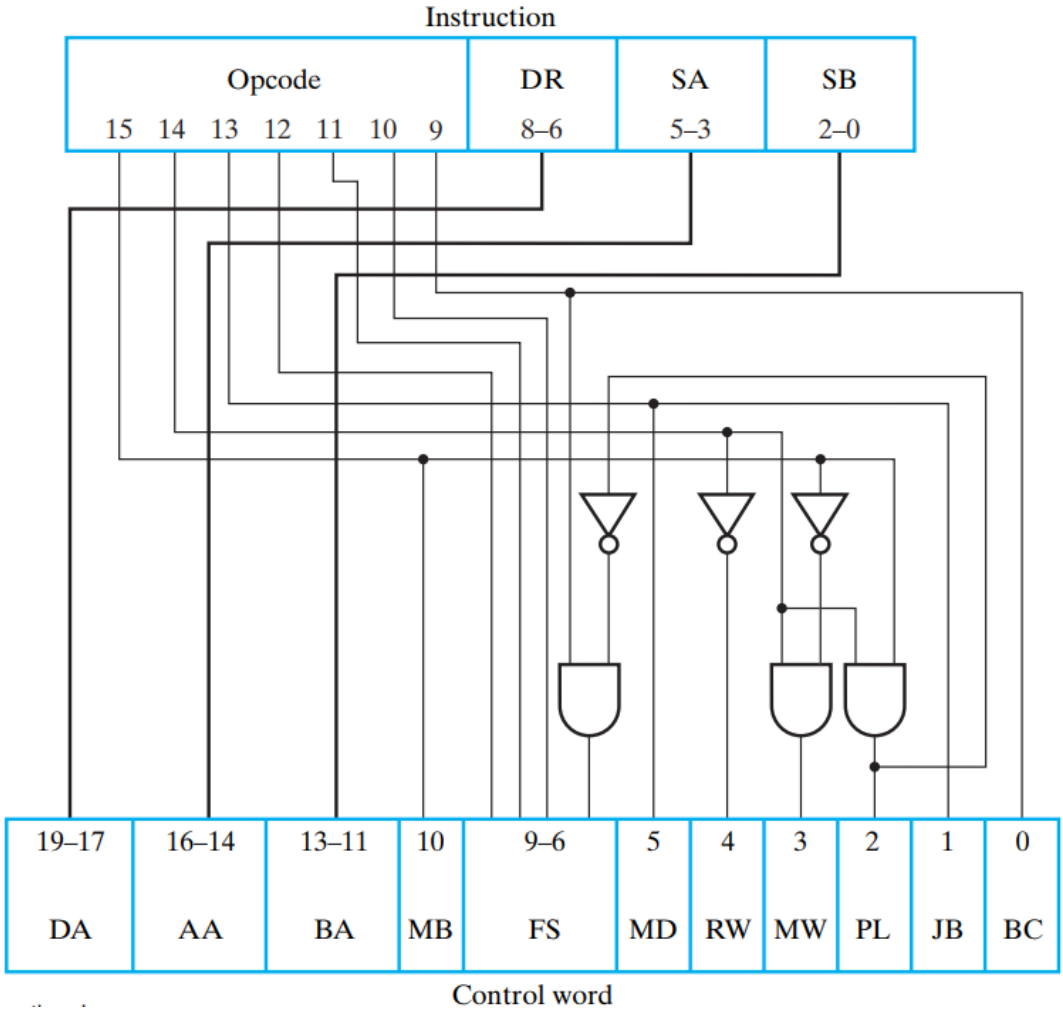

Figure 4 *– Control Unit decoder for the Simple Single Cycle CPU*

1. Implement the instruction decoder in Figure 4 in STRUCTURAL Verilog
   20% (10% for a working design, 10% for proper structural Verilog implementation)
2. Implement the Single-Cycle Computer in Figure 1 in Verilog
   All of the blocks (modules) needed are implemented in this file.
   60% (30% for instantiating the proper modules and making connections, 30% for functionality)
3. Add the following instruction sequence to program memory (ROM).

```
SUB R3, R4, R5
SUB R6, R7, R0
ADD R0, R0, R3
SUB R0, R0, R6
ST  R7, R0
LD  R7, R6
ADI R0, R6, 2
ADI R3, R6, 3
```

   20% (2.5% per correct instruction word)

A testbench for the single cycle processor is provided in single_cycle_CPU_tb.v

You are not required to submit any testbench results; however, you are STRONGLY encouraged to run the testbench to verify the functionality of your processor

Submit your completed single_cycle_CPU.v file to your instructor via slack by 5/5/2018 09:00.