King Fahd University of Petroleum and Minerals
Department of Computer Engineering

COMPUTER ARCHITECTURE COE 308

EXAM II

Problems	Grading
1	
2	
3	
4	
TOTAL	

QUESTION 1: COMPUTER ARITHMETICS

Refer to the Single Precision Floating-Point (SPFP) representation known as the IEEE 754 Floating point standard representation. Answer each of the following questions:

a) In IEEE 754 notation, SPFP numbers are stored as (S, E, M) where S is 1-bit sign, E is 8-bit exponent, and M is 23-bit significand (fraction). Express the corresponding real SPDP number X as function of S, E, and M and the bias.

b) Since the unsigned 8-bit exponent is in the range of (1, 254), determine the real values in the decimal system of the minimum and maximum valid floating-point numbers:

1. The minimum is (, ,) in stored binary and its real decimal value is Amin=

2. The maximum is (, ,) in stored binary and its real decimal value is Amax=

c) Some SPFP arithmetic operations produced the following data (Case 1 to 4):

1. Case-3: (0, 1111 1111, 000 0000 0000 0000 0000 0000):_____
2. Case-2: (1, 0000 0000, 000 0000 0000 0000 0000 0000): _____
3. Case-4: (0, 1111 1111, 000 0000 0000 0000 0001 0000): _____
4. Case-5: (0, 0000 1111, 000 0000 0000 0000 0001 0000): _____
5. Case-1: (1, 1111 1111, 000 0000 0000 0000 0000 0000): _____

Classify each of the above stored binary (case1 to 5) as either: (1) -infinity, (2) +infinity, (3) valid zero, (4) valid SPDF, and (5) a NAN.

Solution:

a) In IEEE 754 notation, a number that is stored as (S, E, M) where S is one sign bit, E is 8 bit exponent, and M is 23-bit significand. Then the corresponding real SPFP is X = (-1)^S * (1 + M) * 2^(E - 127), where 127 is the used bias.

b) Since the unsigned 8-bit exponent is in the range of (1, 254), the valid minimum and maximum floating-point numbers are (sign-1, Exp-8, Man-23):

1. The minimum is (0, 0000 0001, 000 0000 0000 0000 0000 0000) as stored binary and its decimal value is Amin= (-1)^0 * 1.0 * 2^{1-127}= 1.0 * 2^{-126}= 1.175*10^-38

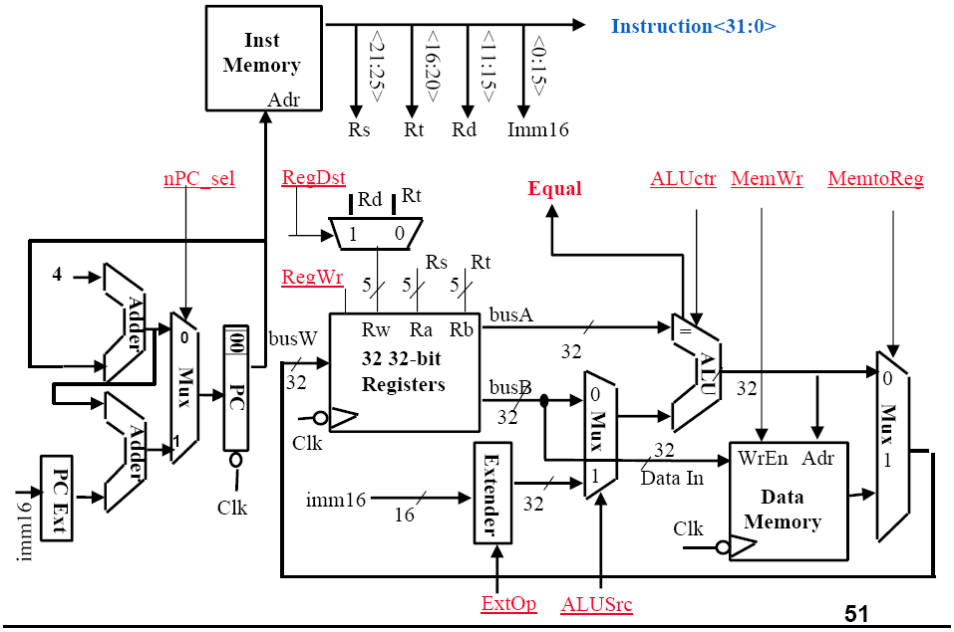
2. The maximum is (0, 1111 1110, 111 1111 1111 1111 1111 1111) as stored binary and its decimal value is Amax = (-1)^0 * (1.0 + 1.0 – 2^(-23)) * 2^{254-127}= (2.0-2^(-23)) * 2^{127}=3.4*10^38

c) Some SPFP arithmetic operations produced the following data (Case 1 to 4):

1. Case-3: (0, 1111 1111, 000 0000 0000 0000 0000 0000), which is +infinity
2. Case-2: (1, 0000 0000, 000 0000 0000 0000 0000 0000), which is zero
3. Case-4: (0, 1111 1111, 000 0000 0000 0000 0001 0000), which is a NAN
4. Case-5: (0, 0000 1111, 000 0000 0000 0000 0001 0000), which is valid SPFP
5. Case-1: (1, 1111 1111, 000 0000 0000 0000 0000 0000), which is - infinity

QUESTION 2: ANALYSIS OF SINGLE CYCLE DATAPATH

Consider the MIPS Instruction set and the following **Single-Cycle Datapath** (**SCDP**) as one alternative to MIPS processor design:



Answer each of the following questions:

1. For what reason some components of the above **SCDP** are clocked! Provide justification to the clocking of each such a component by referring to the MIPS instruction set (IS) whenever needed.

2. For what reason a clock control is needed only for write operations. Is this clock control sufficient to ensure proper functioning of all the write operations!

3. Fill-in the control values as logic 0, logic 1 or a Don't Care (X) for each of the listed control and each of the listed instructions:

	add	sub	ori	lw	sw	beq	jump
RegDst							
ALUSrc							
MemtoReg							
RegWrite							
MemWrite							
nPCsel							
Jump							
ExtOp							

4. Design of the PC updating hardware by using minimum possible hardware.
Solution:

1. Any instruction produces a writable result at its last step. This leads all write operations to be completed (synchronized) with the falling edge at the end of the clock For this: (1) the PC is updated to provide the address of the next instruction, (2) the write operation to Register File and Data-memory are both clocked using the falling edge at the end of the clock
2. The effect of Read operation can be undone, however, the write (destructive) the operation cannot be undone. Also the write occurs at end of cycle as the result of some instruction execution. Therefore only the writer operations (PC,RF, and D-Mem) are clocked. The clock control is not sufficient because each instruction leads some write to take place. The specific write must be enabled depending on the instruction.
3. The control values are filled below as logic 0, logic 1 or a Don't Care (X) for each of the listed control and each of the listed instructions (ignoring the last row for the ALUcntr):

See Appendix A	func	10 0000	10 0010	We Don't Care :-)				
	op	00 0000	00 0000	00 1101	10 0011	10 1011	00 0100	00 0010
		add	**sub**	**ori**	**lw**	**sw**	**beq**	**jump**
RegDst		1	1	0	0	x	x	x
ALUSrc		0	0	1	1	1	0	x
MemtoReg		0	0	0	1	x	x	x
RegWrite		1	1	1	1	0	0	0
MemWrite		0	0	0	0	1	0	0
nPCsel		0	0	0	0	0	1	0
Jump		0	0	0	0	0	0	1
ExtOp		x	x	0	1	1	x	x
ALUctr<2:0>		Add	Subtract	Or	Add	Add	Subtract	xxx

4. The three cases for updating the Program Counter (PC) are:

   - PC +4 for all instruction except for B-type and Beq when the branch is not Taken,
   - PC + 4 + Sing-Ext(32-bit)(Left-Shift(2)(Immediate-16)) which is Sign-Extended(32-bit) of Imm-16 after a Left-shift(2) which useful for BEQ when the branch is Taken,
   - A 32-bit formed by (MSB(4)(PC +4), Immediate-26-bit, 00), where the 4 Msb of PC+4 are concatenated to (Immediate-26-bit) after adding 2 zeros at the Lsb position. This is needed for the unconditional branching or B-type such as instruction J.

One MUX is controlled by nPCsel is used to select the first or the second option nPCsel= (JEQ and Z) which is 1 for a JEQ if the Zero flag is on, and zero otherwise. A second MUX is used to select the output of previous MUX or the 3rd option using control C= J.