$F + \bar{G} + 1 = F - G$

Faculty of Computers and Artificial Intelligence
Cairo University

Final Exam

Program:	**Computer Science / Software Engineering**	**Date:**	**15/1/2022**
Course Name:	**Computer Organization and Architecture**	**Duration:**	**2 hours**
Course Code:	**CS331**	**Total Marks:**	**60 marks**
Instructor(s):	**Dr. Amin Allam**		

تعليمات هامة:

• حيازة التليفون المحمول مفتوحا داخل لجنة الأمتحان يعتبر حالة غش تستوجب العقاب وإذا كان ضرورى الدخول بالمحمول فيوضع مغلقا فى الحقائب .
• لا يسمح بدخول سماعة الأذن أو البلوتوث .
• لايسمح بدخول أي كتب أو ملازم أو أوراق داخل اللجنة والمخالفة تعتبر حالة غش .

- **Exam consists of 40 multiple-choice questions in 6 pages. Each question weights 1.5 marks.**
- **Record in the bubble sheet exactly ONE answer for each question.**

Qa ⇒ A 4×1 multiplexer with selection inputs S_1 and S_0 selects input 0 when $S_1S_0 = 00$, selects input 1 when $S_1S_0 = 01$, selects input 2 when $S_1S_0 = 10$, and selects input 3 when $S_1S_0 = 11$.
For questions 1 to 6, consider an electronic circuit which has:
- Inputs: S_1, S_0, C_0, the signed binary integer $F = F_1F_0$, and the signed binary integer $G = G_1G_0$.
- output: the signed binary integer $V = V_1V_0$. (All signed integers are in the 2's complement form).

The circuit consists of the following items:
- A 4×1 multiplexer with selection inputs S_1, S_0, and 4 inputs: Q_0, Q_1, Q_2, Q_3, and output: M_0.
- Another 4×1 multiplexer with selection inputs S_1, S_0, and 4 inputs: R_0, R_1, R_2, R_3, and output: M_1.
- A full-adder with inputs: F_0, M_0, C_0 and outputs: C_1 (carry), V_0 (sum).
- Another full-adder with inputs: F_1, M_1, C_1 and outputs: C_2 (carry), V_1 (sum).
- $\overline{G_0}$ is connected to Q_0. • $\overline{G_1}$ is connected to R_0. • G_0 is connected to Q_1. • G_1 is connected to R_1.
- Logic 0 is connected to both Q_2 and R_2. • Logic 1 is connected to both Q_3 and R_3.

1 After $S_1 = 0$, $S_0 = 0$, $C_0 = 0$, the output V will equal to: $F - G - 1 : F - G$
A $F+G$ B $F-G$ C $F+G+1$ [✓ D] $F-G-1$ E $F-G+1$

2 After $S_1 = 0$, $S_0 = 0$, $C_0 = 1$, the output V will equal to: $F + \bar{G}$
A $F+G$ [✓ B] $F-G$ C $F+G+1$ D $F-G-1$ E $F-G+1$

3 After $S_1 = 0$, $S_0 = 1$, $C_0 = 1$, the output V will equal to: G
A $F+G$ B $F-G$ [✓ C] $F+G+1$ D $F-G-1$ E $F-G+1$

4 After $S_1 = 1$, $S_0 = 0$, $C_0 = 1$, the output V will equal to: 0 17.8
A $F+G$ B $F-G$ [✓ C] $F+1$ D $F-1$ E F

5 After $S_1 = 1$, $S_0 = 1$, $C_0 = 0$, the output V will equal to:
A $F+G$ B $F-G$ C $F+1$ [✓ D] $F-1$ E F

6 After $S_1 = 1$, $S_0 = 1$, $C_0 = 1$, the output V will equal to: 1
A $F+G$ B $F-G$ C $F+1$ D $F-1$ [✓ E] F

3 3 3 3 2 3
B B B C C A+
٣ ٣ ٣ 2.4 2.4 ٤.4

Qb

7 The outputs of 32 registers are connected to the inputs of a bus. Each register stores 128 bits. The bus selects the outputs of one of the input registers according to the selection inputs of the bus. The bus must have the following number of **selection inputs**:

A 4 B 5 [✓ C] 7 D 32 E 128

8 The outputs of 32 registers are connected to the inputs of a bus. Each register stores 7 bits. The bus selects the outputs of one of the input registers according to the selection inputs of the bus. The bus must have the following number of **multiplexers**:

A 4 B 5 [✓ C] 7 D 32 E 128

9 A memory unit contains 1024 words. Each word is 32 bits. The number of **input address lines** to this memory unit is:

A 5 [✓ B] 10 C 11 D 16 E 32

10 A memory unit contains 1024 words. Each word is 16 bits. The number of **data output lines** out of this memory unit is:

A 5 B 10 C 11 [✓ D] 16 E 32

11 After applying **logical shift right** to a 6-bit register containing the binary number 101100 it becomes:

[✓ A] 010110 B 110110 C 011001 D 011000 E 111100

12 Comparing two numbers to know whether they are equal or not equal, can be done by:

A bitwise complement B bitwise and C bitwise or [✓ D] bitwise xor E arithmetic shift

13 Consider the ADD instruction which increases the value of the accumulator register by the value located at the effective address. If the instruction uses **indirect** addressing and it contains the address 10 in its address field. The memory word at address 10 contains the value 30, and the memory word at address 30 contains the value 45. After the instruction is executed, the accumulator register increases by:

A 10 B 30 C 40 [✓ D] 45 E 75

14 Consider the ADD instruction which increases the value of the accumulator register by the value located at the effective address. If the instruction uses **direct** addressing and it contains the address 10 in its address field. The memory word at address 10 contains the value 30, and the memory word at address 30 contains the value 45. After the instruction is executed, the accumulator register increases by:

A 10 [✓ B] 30 C 40 D 45 E 75

15 The following register contains the instruction that is being executed:

A Program counter B Accumulator [✓ C] Instruction register D Input register E Data register

16 Assume that 4 clock cycles are required on average to execute any instruction. The time required to execute a program containing 10,000 instructions on a computer with a clock rate of 5,000,000 clock cycles per second is:

A 0.002 seconds [✓ B] 0.008 seconds C 125 seconds D 250 seconds E 2000 seconds

⇒ For questions 17 to 20, consider the following control memory of a microprogrammed control unit. Each line represents a microinstruction stored in the control memory.

Microinstruction	Label	Microoperations	Condition	Branch	Address
0	ADD:	NOP	I	CALL	INDRCT
1		READ	U	JMP	NEXT
2		ADD	U	JMP	FETCH
3	INDRCT:	READ	U	JMP	NEXT
4		DRTAR	U	RET	
5	FETCH:	PCTAR	U	JMP	NEXT
6		READ, INCPC	U	JMP	NEXT
7		DRTAR	U	MAP	

17 After microinstruction 0 is executed, the following microinstruction will be executed:
A 1 B 2 [✓ C] 3 D 1 or 3 E unknown

18 After microinstruction 2 is executed, the following microinstruction will be executed:
A 0 B 3 [✓ C] 5 D 3 or 5 E unknown

19 After microinstruction 7 is executed, the following microinstruction will be executed:
A 0 B 4 C 5 D 0 or 4 [✓ E] unknown

20 The following sequence of microinstruction execution is possible:
A 0,1,2,0 B 0,1,2,3 C 0,3,4,0 [✓ D] 0,3,4,1 E 0,3,4,5

Qe ⇒ For questions 21 to 24, consider a binary multiplier which multiplies the multiplicand B=10111 by the multiplier Q=10011 and places the multiplication result into EAQ where A, B, and Q are 5 bits registers and E is a flipflop. Initially, EAQ=00000010011. Each stage starts by performing some necessary operations and completes by shifting right EAQ. The multiplication process completes in 6 stages.

21 After the first stage, EAQ=
A 00000001001 [✓ B] 00101111001 C 00110111001 D 00110011001 E 10110011001

22 After the second stage, EAQ=
A 00010111100 B 00011011100 C 10011001100 D 01011001100 [✓ E] 01000101100

~~23~~ After the third stage, EAQ=
[✓ A] 00100010110 B 00001011110 C 00001101110 D 01001100110 E 00101100110

~~24~~ After the fourth stage, EAQ=
[✓ A] 00010001011 B 00000101111 C 00000110111 D 00100110011 E 00010110011

10111

E	A	Q	SC
0	00000	10011	101
	10111		
0	10111	10011.	
0	01011	11001.	100
	10111		
1	00010	11001	
0	10001	01100	011
0	01000	10110	

Add ~~I~~ A E
Branch I A E

Qd ⇒ For questions 25 to 30, consider a three-segment instruction pipeline I, A, E described as follows. The addressing mode used for the Branch instruction is the register indirect mode:

	Load/Store	Add	Branch
I	Fetch and decode the instruction		
A	Evaluate effective address	Perform the addition	Evaluate branch address
E	Transfer operand from/to memory	Put result in destination register	Put branch address in PC

25 The sequence of instructions: (1) Load R1 (2) Load R2 (3) R3←R1+R2 (4) Store R3 has the following pipeline difficulty:

A Resource conflict [✓ B] Data dependency C Branch D Closed pipeline E Interrupt

26 The pipeline difficulty in the sequence of instructions: (1) Load R1 (2) Load R2 (3) R3←R1+R2 (4) Store R3 can be solved by:

A Adding a no-operation instruction between instructions 1 and 2
B Swapping instructions 2 and 3
[✓ C] Adding a no-operation instruction between instructions 2 and 3
D Swapping instructions 3 and 4
E Adding a no-operation instruction between instructions 3 and 4

27 After solving the pipeline difficulty and pipelining the following sequence of instructions: (1) Load R1 (2) Load R2 (3) R3←R1+R2 (4) Store R3, they will require the following number of clock cycles to be executed:

A 4 B 5 C 6 [✓ D] 7 E 12

28 After solving the pipeline difficulties by inserting no-operation instructions and/or rearranging instructions, such that the resulting pipeline of the following instruction sequence requires the minimum number of clock cycles: (1) Load R1 (2) R2←R1+8 (3) Branch R2 (4) R3←R2+8, they will require the following number of clock cycles to be executed:

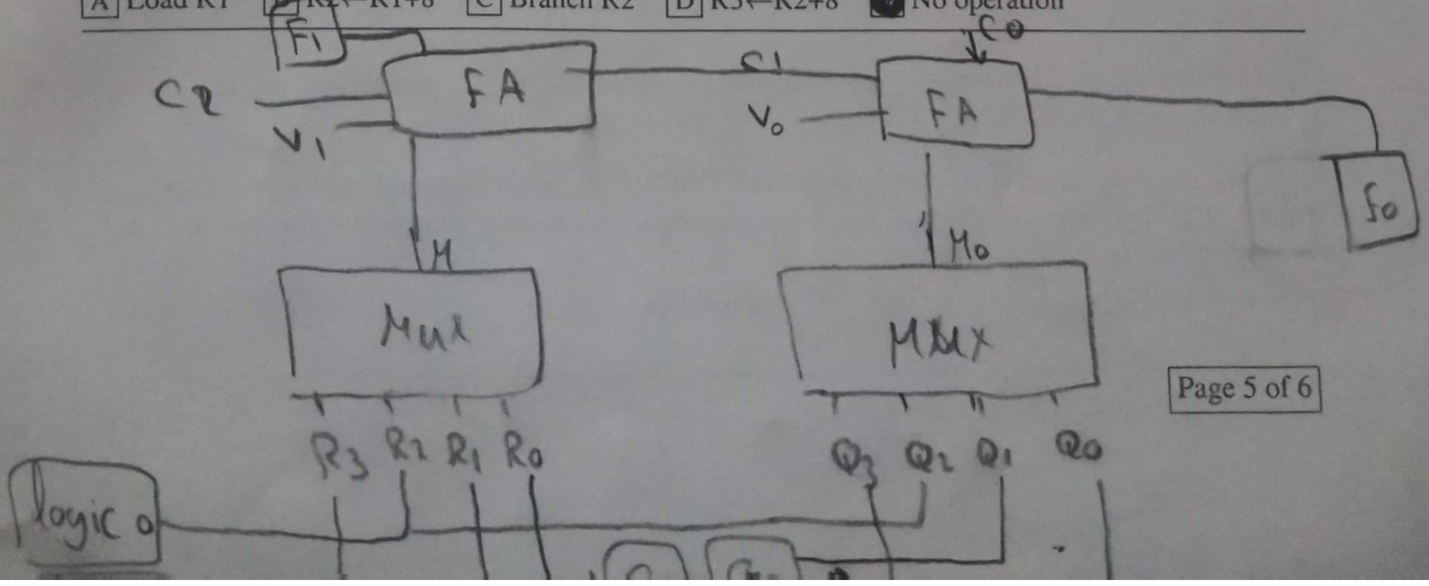
A 5 B 6 C 7 [✓ D] 8 E 12

29 After solving the pipeline difficulties by inserting no-operation instructions and/or rearranging instructions, such that the resulting pipeline of the following instruction sequence requires the minimum number of clock cycles: (1) Load R1 (2) R2←R1+8 (3) Branch R2 (4) R3←R2+8, the third instruction will be:

A Load R1 B R2←R1+8 [✓ C] Branch R2 D R3←R2+8 E No operation

30 After solving the pipeline difficulties by inserting no-operation instructions and/or rearranging instructions, such that the resulting pipeline of the following instruction sequence requires the minimum number of clock cycles: (1) Load R1 (2) R2←R1+8 (3) Branch R2 (4) R3←R2+8, the fourth instruction will be:

A Load R1 ~~B R2←R1+8~~ C Branch R2 D R3←R2+8 [✓ E] No operation



Qe

31 Consider the ADD instruction which increases the value of the accumulator register by the value located at the effective address. If the instruction uses **register** addressing mode and it contains the register R1 in its address field. The register R1 contains the value 10. The memory word at address 10 contains the value 30, and the memory word at address 30 contains the value 45. After the instruction is executed, the accumulator register increases by:

[✓ A] 10 B 30 C 40 D 45 E 75

32 Consider the ADD instruction which increases the value of the accumulator register by the value located at the effective address. If the instruction uses **register indirect** addressing mode and it contains the register R1 in its address field. The register R1 contains the value 10. The memory word at address 10 contains the value 30, and the memory word at address 30 contains the value 45. After the instruction is executed, the accumulator register increases by:

A 10 [✓ B] 30 C 40 D 45 E 75

Push M[A]

33 Let M[X]=the integer value at memory address X. Consider zero-address instructions: Pu
(1) **Push X**: pushes M[X] to stack. (2) **Pop X**: pops the top stack element to M[X].
(3) **Add**: pops the two elements on the top of the stack and pushes their sum.
The minimum number of such instructions that compute M[X]=M[A]+M[B]+M[C] is:

A 3 B 4 [✓ C] 5 D 6 E 7

AC→M[A]

AC→AC+M[B]

34 Let M[X]=the integer value at memory address X. Consider one-address instructions:
(1) **Load X**: sets AC to M[X]. (2) **Store X**: sets M[X] to AC. (3) **Add X**: increases AC by M[X].
The minimum number of such instructions that compute M[X]=M[A]+M[B]+M[C] is:

AC→AC+M[C]

A 3 [✓ B] 4 C 5 D 6 E 7

M[x]?

35 The following event should cause an **internal** interrupt:

A keyboard moved a character to input register B printer ready to take character from CPU
C switching from user mode to administrative mode [✓ D] division by zero E power failure

36 The Zero flipflop becomes 1 if the last operation:

A took a zero operand B result was zero C was no-operation instruction
D took no operands [✓ E] compared two different integers

37 A characteristic of RISC is:

A large number of instructions B variable-length instruction format [✓ C] hardwired control unit
D large variety of addressing modes E small size of control memory

38 The following step is **not** required in multiplication of floating point numbers:

A check for zeros B add the exponents [✓ C] align the mantissas D multiply the mantissas
E normalize the product

10•10

39 The following associative memory word matches **Argument**=10110011 and **Key**=11011000:

[✓ A] 10010111 B 11100111 C 11110111 D 10111111 E 11111111

40 A main memory contains 2^{15} words. Each word is 9 bits. A direct mapping cache memory contains 64 words. The value at address $(34567)_8$ in main memory (8 means octal number) may be cached at the following address in the cache memory:

A $(34)_8$ B $(345)_8$ C $(456)_8$ [✓ D] $(567)_8$ E $(67)_8$

215