Coláiste na Tríonóide, Baile Átha Cliath
Trinity College Dublin
Ollscoil Átha Cliath | The University of Dublin

ering, Mathematics and Science

mputer Science & Statistics

Integrated Computer Science Programme **Michaelmas Term 2022**
Year 2

CSU22022 – Computer Architecture I

31. August 2022 at 14.00 - 19.00 **5-hour take-home exam**

Prof. Michael Manzke

Answer Question 1 and 2. Please confirm in you answer that this is your own work and that you have not collaborated with other students.

Students who are registered with Disability Services and who are entitled to extra time in examinations will be granted 10 additional minutes per hour (i.e. a 50 minutes period in which to complete the 5-hour take home exam).

Figure 1 depicts the processor we designed. You will need this diagram for all the questions.

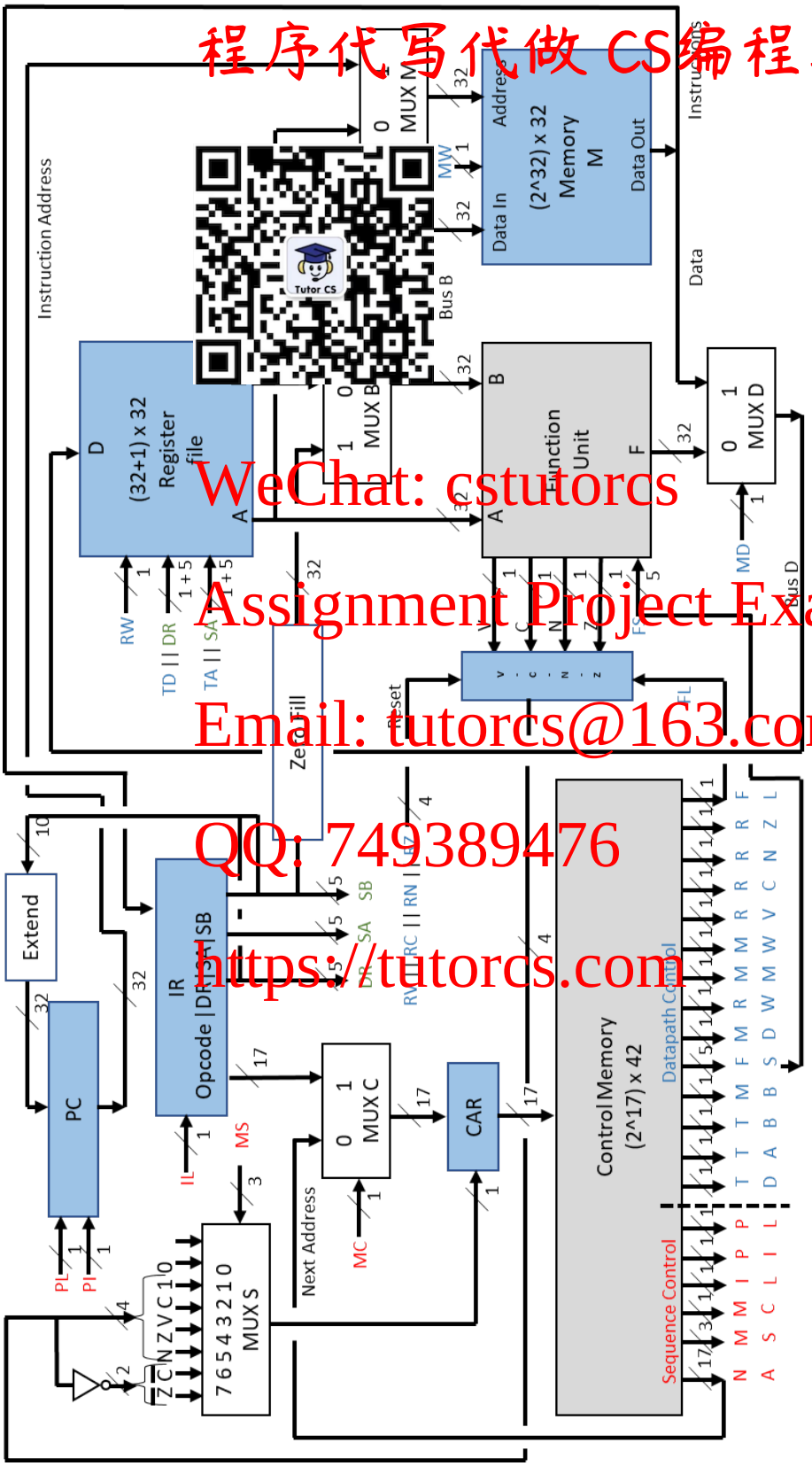


Figure 1 Processor block diagram

Figure 2 shows various instruction formats. These 32bit instructions (machine-code) are stored in the **Memory M** (see Figure 1). The **PC** (Program Counter, see Figure 1) should point to the next to be executed instruction in **Memory M**. You require this information to program the **machine-code** in **Memory M** (see Figure 1).

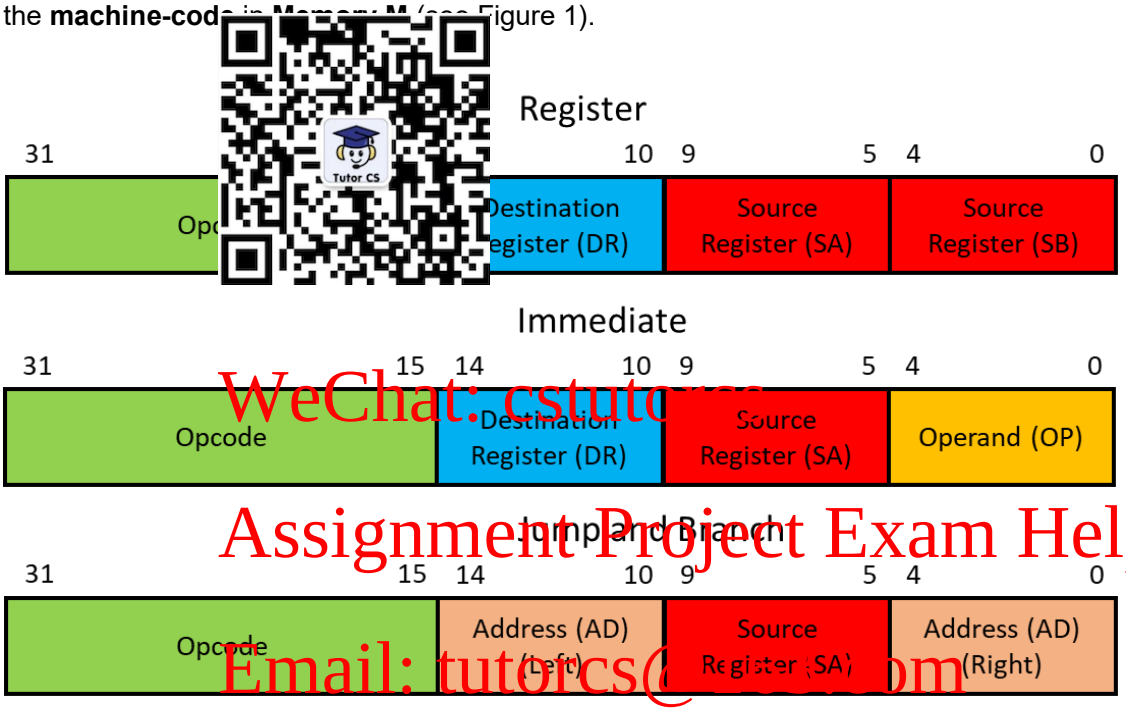


Figure 2

Figure 3 provides you with the binary code for all the operations of the **Function Unit** (see Figure 1). You need this information to program the **micro-code** in the **Control Memory** (see Figure 1).

Table 1: FS code definition

FS	MF Select	G Select	H Select	Micro-operation
00000			00	$F = A$
00001			00	$F = A + 1$
00010			00	$F = A + B$
00011			00	$F = A + B + 1$
00100			01	$F = A + \bar{B}$
00101			01	$F = A + \bar{B} + 1$
00110	0	0110	01	$F = A - 1$
00111	0	0111	01	$F = A$
01000	0	1000	00	$F = A \wedge B$
01010	0	1010	10	$F = A \vee B$
01100	0	1100	10	$F = A \oplus B$
01110	0	1110	10	$F = \bar{A}$
10000	1	0000	00	$F = sr$
10100	1	0100	01	$F = srB$
11000	1	1000	10	$F = slB$

Figure 3

Figure 4 depicts the layout of all the memory locations in the in the **Control Memory** (see Figure 1). You need this information to program the micro-code in the **Control Memory** (see Figure 1). Figure 3 provides with all the information for programming bit 9 to 13.

41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Next Address																				MS		MC	IL	PI	PL	TD	TA	TB	MB	FS				MD	RW	MM	MW	RV	RC	RN	RZ	FL

Figure 4

1. **Question**, please provide an algorithmic state machine chart for the implementation of the following **machine-code instructions**; ***ADI***, ***LD***, ***ST***, ***INC***, ***NOT***, and ***ADD***. Please provide **micro-code** at the correct memory location in the **Control Memory** and **machine-code instructions** at the correct memory location in the **Memory M**.

IMPORTANT: ADI, LD, ST, INC, NOT, and ADD instructions in the Control Memory are determined by the last digit of your student number (ID). The micro-code for the ADI, LD, ST, INC, NOT, and ADD instructions must be in consecutive memory locations and determined by the last digit of your student number (ID). Your 1st micro-code address (Start Address) of your 1st instruction is determined by the last two digits of your student number (ID) **e.g. ID 263357(25) = ??? ???? = 001 1001**. Table 1 provides the correct micro-code order for the last digit of your student number (ID) e.g. **ID 2633572(5) = INC, ADI, NOT, ADD, ST, and LD**. The micro-code for the IF (Instruction Fetch) and the EXO can be placed anywhere in the control memory.

For the example ID 2633726:

1st 0 0000 0000 0001 1001 (**INC** micro-code)

2nd 0 0000 0000 0001 1010 (**ADI** micro-code)

3rd 0 0000 0000 0001 1011 (**NOT** micro-code)

4th 0 0000 0000 0001 1100 (**ADD** micro-code)

5th 0 0000 0000 0001 1101 (**ST** micro-code)

6th 0 0000 0000 0001 1110 (**LD** micro-code)

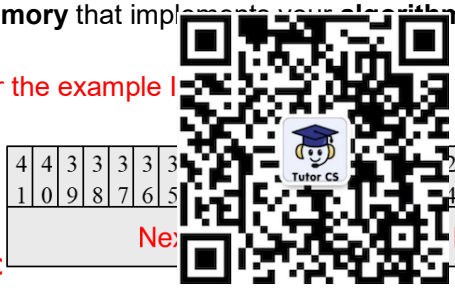
You must provide an algorithmic state machine chart for your micro-code that includes the micro-code for the IF (Instruction Fetch) and the EXO.

ID	Start Address	1st	2nd	3rd	4th	5th	6th
0	0 0000 0000 0??? ????	LD	NOT	INC	ST	ADD	ADI
1	0 0000 0000 0??? ????	ADD	ST	NOT	ADI	INC	LD
2	0 0000 0000 0??? ????	ADD	ST	ADI	NOT	LD	INC
3	0 0000 0000 0??? ????	LD	NOT	ADI	ADD	INC	ST
4	0 0000 0000 0??? ????	ST	INC	LD	NOT	ADD	ADI
5	0 0000 0000 0??? ????	INC	ADI	NOT	ADD	ST	LD
6	0 0000 0000 0??? ????	INC	NOT	LD	ADI	ADD	ST
7	0 0000 0000 0??? ????	NOT	ST	ADI	LD	ADD	INC
8	0 0000 0000 0??? ????	ADD	ADI	INC	ST	LD	NOT
9	0 0000 0000 0??? ????	ST	INC	ADD	LD	ADI	NOT

Table 1

Please see below for the correct layout of your answer. Your answer must provide **micro-code** for the **Control Memory** that implements your **algorithmic state machine chart.** The **micro-code** should have the following format for every memory address in the **Control Memory** that implements your **algorithmic state machine**.

For the example I



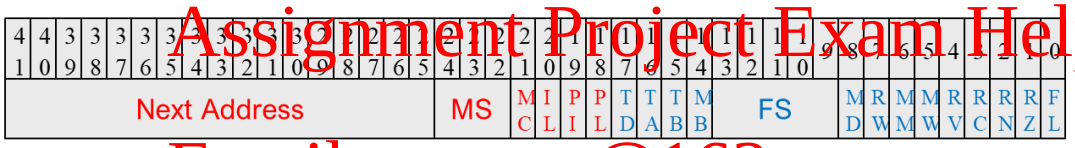
41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Nex																	MS			MC	IL	PI	PL	TD	TA	TB	MB	FS					MD	RW	MM	MW	RV	RC	RN	RZ	FL

INC

a) Control Memory Address = 0 0000 0000 0001 1001

b) Binary code for bits 0 to 41

c) Providing written reasons for selecting these binary values for NA, MS, MC, IL, PI, PL, TD, TA, TB, MB, FS, MD, RW, MM, and MW.



41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Next Address																	MS			MC	IL	PI	PL	TD	TA	TB	MB	FS					MD	RW	MM	MW	RV	RC	RN	RZ	FL

ADI

a) Control Memory Address = 0 0000 0000 0001 1010

b) Binary code for bits 0 to 41

c) Providing written reasons for selecting these binary values for NA, MS, MC, IL, PI, PL, TD, TA, TB, MB, FS, MD, RW, MM, and MW.

… continue for all states of your algorithmic state machine and don't forget the micro-code for the IF (Instruction Fetch) and the EX0.

[40 marks]

Your solution must also provide **machine-code instructions** at the correct memory location in the **Memory M**. The **machine-code instructions** should have the following format. Also, you should execute the ***machine-code instructions*** in the following order ***ADD***, ***INC***, ***NOT***, ***ADI***, ***ST***, and ***LD***. Your [illegible]structions address (***ADD***) is determined by the last digit of your student nu[illegible]35725 = 0000 0000 0000 0101

For example:

31 Opcode 10	9 5	4 0

ADD

a) Memory M Address (for the example ID 26335725 = 0000 0000 0000 0101)

b) Binary code for bits 0 to 31

c) Providing written reasons for selecting these binary values for bits 0 to 31

31 Opcode 15 14	10 9	5 4	0

INC

a) Memory M Address (for the example ID 26335725 = 0000 0000 0000 0110)

b) Binary code for bits 0 to 31

c) Providing written reasons for selecting these binary values for bits 0 to 31

… continue for all **machine-code instructions** (***NOT***, ***ADI***, ***ST***, and ***LD***)

You should assume the following values for the six **machine-code instructions** (***ADD***, ***INC***, ***NOT***, ***ADI***, ***ST***, and ***LD***)

a) ***ADD***: DR=0 1001, SA=0 1010, SB=0 1011

b) ***INC***: DR=01010, SA=0 0110

c) ***NOT***: DR=0 0111, SA=0 1000

d) ***ADI***: DR=0 0000, SA=0 0001, zfIR[4:0]=0 0010

e) ***ST***: SA=0 0101, SB=110

f) ***LD***: DR=0 0011, SA=0 0100

[20 marks]

2. **Question**, this question builds on Question 1. You must modify your **algorithmic state machine chart** from Question 1 by incorporating the **algorithmic state machine chart** shown in Figure 5 (below) into your ASM chart.

Please provide **micro-code** at the correct memory location in the **Control Memory** and a **machine-code instruction (LRI)** at the correct memory location in the **Memory M** that will invoke the **micro-code instructions**.

You only need to provide **micro-code** at the correct memory location in the **Control Memory** for **those that implement the LRI instruction (Figure 5)**

Please follow the **micro-code** and **machine-code instructions**. Please see below Figure 5 for the correct layout of you answer.

IMPORTANT: The first micro-code for the LRI instruction (R32<-M[R[SA]]) must be after your last micro-code address from Question 1, e.g. for ID 26335725, the last micro-code address is 6th 0 0000 0000 0001 1110 (LD micro-code). Therefore, a student with this ID must implement the micro-code for R32<-M[R[SA]] at the next memory location in the control memory 0 0000 0000 0001 1111. All four operations must be implemented in consecutive memory locations in the control memory.

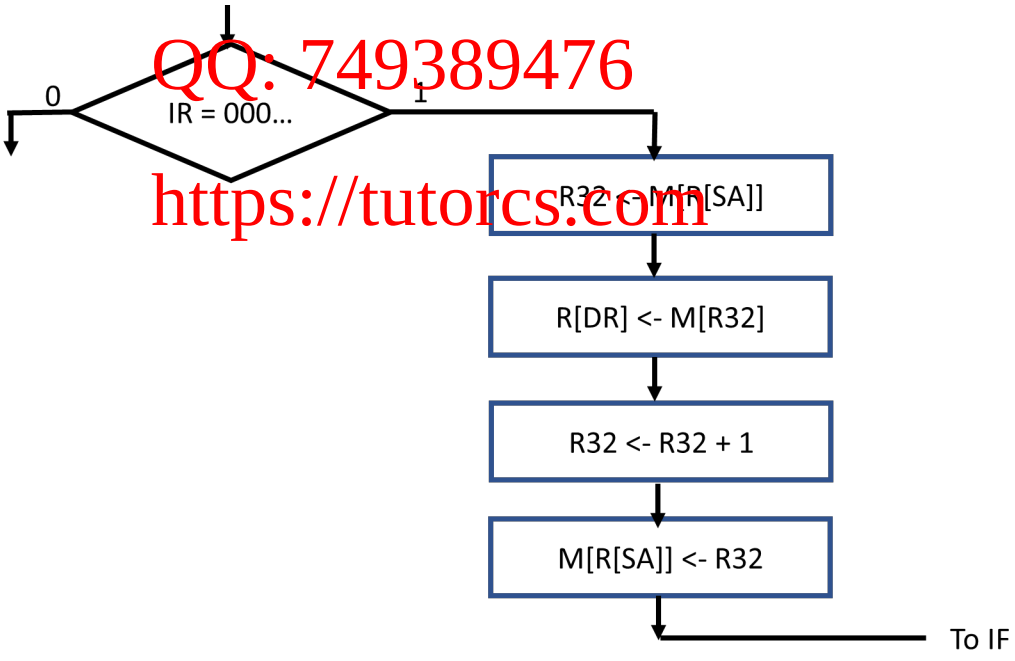
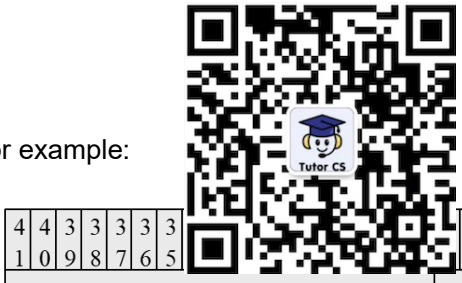


Figure 5

Your answer must provide **micro-code** for the **Control Memory** that implements the *modified* **algorithmic state machine chart.** The **micro-code** should have the following format for memory addresses in the **Control Memory** that implements your **algorithmic state machine: IF, EXO, and those that implement the LRI instruction.**

For example:



	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1st	Next Address																		MS		MC	IL	PI	PL	TD	TA	TB	MB	FS					MD	RW	MM	MW	RV	RC	RN	RZ	FL
2nd	Next Address																		MS		MC	IL	PI	PL	TD	TA	TB	MB	FS					MD	RW	MM	MW	RV	RC	RN	RZ	FL

a) Control Memory Address

b) Binary code for bits 0 to 41

c) Providing written reasons for selecting these binary values for NA, MS, MC, IL, PI, PL, TD, TA, TB, MB, FS, MD, RW, MM, and MW.

… continue for all states of your algorithmic state machine.

[35 marks]

Your solution must also provide a **LRI machine-code instruction** at the correct memory location in the **Memory M**. The **machine-code instruction** should have the following format.

You should execute this ***machine-code instruction*** *after* the ***LD* machine-code instruction** (1. Question).

31 Opcode	10	9 5	4 0

LRI

a) Memory M Adc

b) Binary code for bits 0 to 31

c) Providing written reasons for selecting these binary values for bits 0 to 31

You should assume the following values for the ***LRI* machine-code instruction**:

a) ***LRI***: DR=01110, SA=11011

[5 marks]

IMPORTANT ! : You may provide a hand written or electronic solution