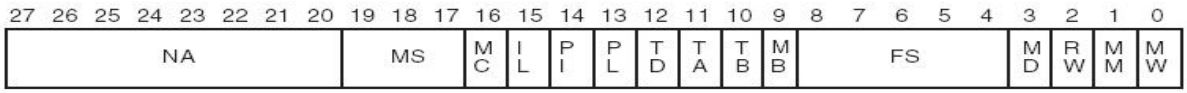
Computer Architecture CSU22022 Exam 2020
Tanmay Kaushik
Student No. – 18308341
Email ID: kaushikt@tcd.ie

Question 1
Micro-Code



ADI

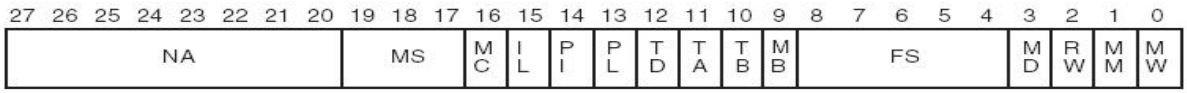
a) **Control Memory Address:** 00000000
b)

NA	MS	MC	IL	PI	PL	TD	TA	TB	MB	FS	MD	RW	MM	MW
11000000	001	0	0	0	0	0	0	0	1	00010	0	1	0	0

c) Reasons-

- **a. NA : 11000000**
  - The reason for this is the address of the IF block which is the next block according to the ASM diagram has the address 11000000.
- **b. MS : 001**
  - To increment the Control Address Register, we select MS as 001
- **c. MC : 0**
  - The loaded address in Instruction Register does not have to be used in the next CAR. Therefore, MC is 0
- **d. IL : 0**
  - The instruction is not fetched by Memory. So IL is 0.
- **e. PI : 0**
  - PI is not set as program counter need not be incremented.
- **f. PL : 0**
  - As PC doesn't load other memory so, PC is set as 0.
- **g. TD : 0**
- **h. TA : 0**
- **i. TB : 0**
  - The reason that TD,TA,TB are 0 because R8 – which is the extra register - is not used.
- **j. MB : 1**
  - As we need an operand, and based on figure 3 we select MD as 1
- **k. FS : 00010**
  - The is derived from Figure 3 as ADI function is written as A + B.
- **l. MD : 0**
  - MD is set as 0 to select function unit instead on Data Input from Register File
- **m. RW : 1**
  - As we need to store the new value RW(Read 0 / Write 1) is 1
- **n. MM : 0**
  - MM is set as 0 to as we don't select Program Counter.
- **o. MW : 0**

- We don't write value to selected memory address so MW is set as 0.



LD

a) **Control Memory Address:** 00000001
b)

NA	MS	MC	IL	PI	PL	TD	TA	TB	MB	FS	MD	RW	MM	MW
11000000	**001**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**00000**	**1**	**1**	**0**	**0**

c) Binary Values along with reasons-

a. NA : 11000000
- The reason for this is the address of the IF block which is the next block according to the ASM diagram has the address 11000000.

b. MS : 001
- To increment CAR register we ser MS to 001

c. MC : 0
- The loaded address in Instruction Register does not have to be used in the next CAR. Therefore, MC is 0

d. **IL : 0**
- The instruction is not fetched by Memory. So IL is 0.

e. PI : 0
- PI is 0 as Program Counter is unchanged.

f. PL : 0
- PL is 0 as Program Counter is unchanged.

g. TD : 0
h. TA : 0
i. TB : 0
- The reason that TD,TA,TB are 0 because R8 – which is the temp register - is not used.

j. MB : 0
- No additional operand is used so MB is 0.

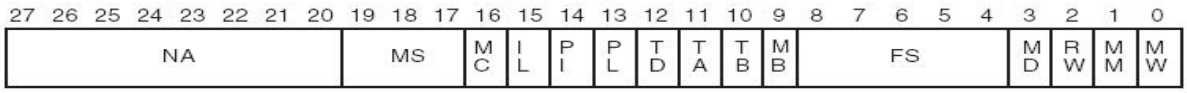
k. FS : 00000
- As it is similar to F = A, hence FS = 00000 referring to figure 3.

l. MD : 1
- MD is set as 1 as we take in Data.

m. RW : 1
- As we are writing data so RW(Read 0 / Write 1) is set as 1.

n. MM : 0
- Reg A is used for input so MM is set to 0

o. MW : 0
- MW is set as 0 as we don't write to memory.



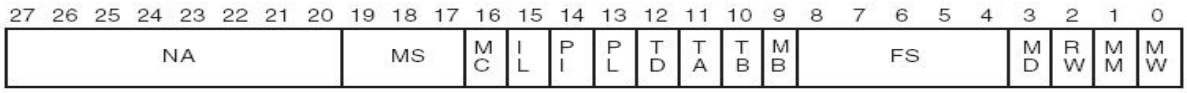
ST

a) **Control Memory Address:** 00000010
b)

NA	MS	MC	IL	PI	PL	TD	TA	TB	MB	FS	MD	RW	MM	MW
11000000	001	0	0	0	0	0	0	0	0	00000	0	0	0	1

c) Binary Values along with reasons-

- **a. NA : 11000000**
  - The reason for this is the address of the IF block which is the next block according to the ASM diagram has the address 11000000.
- **b. MS : 001**
  - To increment CAR register we ser MS to 001
- **c. MC : 0**
  - The loaded address in Instruction Register does not have to be used in the next CAR. Therefore, MC is 0
- d. **IL : 0**
  - The instruction is not fetched by Memory. So IL is 0.
- **e. PI : 0**
  - PI is 0 as Program Counter is unchanged.
- **f. PL : 0**
  - PL is 0 as Program Counter is unchanged.
- **g. TD : 0**
- **h. TA : 0**
- **i. TB : 0**
  - The reason that TD,TA,TB are 0 because R8 – which is the temp register - is not used.
- **j. MB : 0**
  - No additional operand is used so MB is 0.
- **k. FS : 00000**
  - As it is similar to F = A, hence FS = 00000 referring to figure 3.
- **l. MD : 0**
  - MD is set as 0 as we take select Function Unit instead of taking Data In.
- **m. RW : 0**
  - As we are reading data from the register so RW is set as 0.
- **n. MM : 0**
  - Reg A is used for input so MM is set to 0
- **o. MW : 1**
  - MW is set as 1 as we write to memory.



INC

a) **Control Memory Address:** 00000011
b)

NA	MS	MC	IL	PI	PL	TD	TA	TB	MB	FS	MD	RW	MM	MW
11000000	001	0	0	0	0	0	0	0	0	00001	0	1	0	0

c) Binary Values along with reasons-

a. NA : 11000000
- The reason for this is the address of the IF block which is the next block according to the ASM diagram has the address 11000000.

b. MS : 001
- To increment CAR register we ser MS to 001

c. MC : 0
- The loaded address in Instruction Register does not have to be used in the next CAR. Therefore, MC is 0

d. **IL : 0**
- The instruction is not fetched by Memory. So IL is 0.

e. PI : 0
- PI is 0 as Program Counter is unchanged.

f. PL : 0
- PL is 0 as Program Counter is unchanged.

g. TD : 0

h. TA : 0

i. TB : 0
- The reason that TD,TA,TB are 0 because R8 – which is the temp register - is not used.

j. MB : 0
- No additional operand is used so MB is 0.

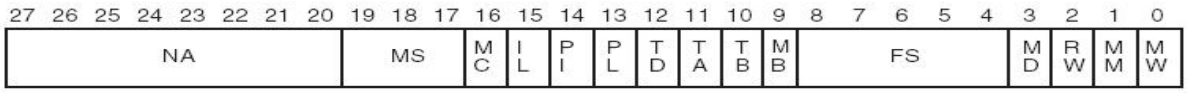
k. FS : 00001
- As it is similar to F = A + 1, hence FS = 00001 referring to figure 3.

l. MD : 0
- MD is set as 0 as we take select Function Unit instead of taking Data In.

m. RW : 1
- As we are writing data to the register so RW is set as 0.

n. MM : 0
- Reg A is used for input so MM is set to 0

o. MW : 0
- MW is set as 0 as we don't write to memory.



NOT

a) **Control Memory Address:** 00000100
b)

NA	MS	MC	IL	PI	PL	TD	TA	TB	MB	FS	MD	RW	MM	MW
11000000	001	0	0	0	0	0	0	0	0	01110	0	1	0	0

c) Binary Values along with reasons-

a. NA : 11000000
- The reason for this is the address of the IF block which is the next block according to the ASM diagram has the address 11000000.

b. MS : 001
- To increment CAR register we ser MS to 001

c. MC : 0
- The loaded address in Instruction Register does not have to be used in the next CAR. Therefore, MC is 0

d. **IL : 0**
- The instruction is not fetched by Memory. So IL is 0.

e. PI : 0
- PI is 0 as Program Counter is unchanged.

f. PL : 0
- PL is 0 as Program Counter is unchanged.

g. TD : 0

h. TA : 0

i. TB : 0
- The reason that TD,TA,TB are 0 because R8 – which is the temp register - is not used.

j. MB : 0
- No additional operand is used so MB is 0.

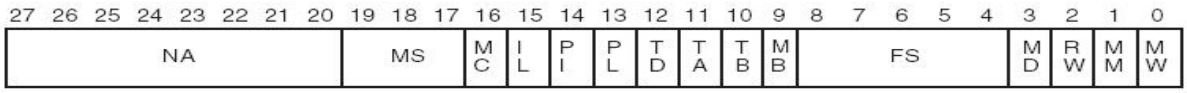
k. FS : 01110
- As it is similar to F = not(A) , hence FS = 01110 referring to figure 3.

l. MD : 0
- MD is set as 0 as we take select Function Unit instead of taking Data In.

m. RW : 1
- As we are writing data to the register so RW is set as 0.

n. MM : 0
- Reg A is used for input so MM is set to 0

o. MW : 0
- MW is set as 0 as we don't write to memory.



ADD

a) **Control Memory Address:** 00000101
b)

NA	MS	MC	IL	PI	PL	TD	TA	TB	MB	FS	MD	RW	MM	MW
11000000	001	0	0	0	0	0	0	0	0	00010	0	1	0	0

c) Binary Values along with reasons-

- **a. NA : 11000000**
  - The reason for this is the address of the IF block which is the next block according to the ASM diagram has the address 11000000.
- **b. MS : 001**
  - To increment CAR register we ser MS to 001
- **c. MC : 0**
  - The loaded address in Instruction Register does not have to be used in the next CAR. Therefore, MC is 0
- d. **IL : 0**
  - The instruction is not fetched by Memory. So IL is 0.
- **e. PI : 0**
  - PI is 0 as Program Counter is unchanged.
- **f. PL : 0**
  - PL is 0 as Program Counter is unchanged.
- **g. TD : 0**
- **h. TA : 0**
- **i. TB : 0**
  - The reason that TD,TA,TB are 0 because R8 – which is the temp register - is not used.
- **j. MB : 0**
  - No additional operand is used so MB is 0.
- **k. FS : 00010**
  - As it is similar to F = A + B , hence FS = 00010 referring to figure 3.
- **l. MD : 0**
  - MD is set as 0 as we take select Function Unit instead of taking Data In.
- **m. RW : 1**
  - As we are writing data to the register so RW is set as 0.
- **n. MM : 0**
  - Reg A is used for input so MM is set to 0
- **o. MW : 0**
  - MW is set as 0 as we don't write to memory.

Machine Code

LD

- **Memory M Address :** 0000 0000 0000 0101
- **Binary code for bits 0 to 15 :** 0000 0001 111 011 000
- Reasons-
    a. **Opcode** : 0000 0001 as this is the address of LD instruction in Control Memory.
    b. **8 to 6** : Store in Reg[7] as DR(=111) = 7
    c. **5 to 3** : Memory Access at Reg[3] as SA(=011) = 3
    d. **2 to 0.** : No Register B

2

INC

- **Memory M Address :** 0000 0000 0000 0110
- **Binary code for bits 0 to 15 :** 0000 0011 010 011 000
- Reasons-
    a. **Opcode** : 0000 0011 as this is the address of INC instruction in Control Memory.
    b. **8 to 6** : Store in Reg[2] as DR(=010) = 2
    c. **5 to 3** : Memory Access at Reg[3] as SA(=011) = 3
    d. **2 to 0.** : No Register B

ADI

- **Memory M Address :** 0000 0000 0000 0111
- **Binary code for bits 0 to 15 :** 0000 0000 001 010 010
- Reasons-
    a. **Opcode** : 0000 0000 as this is the address of ADI instruction in Control Memory.
    b. **8 to 6** : Store in Reg[1] as DR(=001) = 1
    c. **5 to 3** : Memory Access at Reg[2] as SA(=010) = 2
    d. **2 to 0.** : Integer SB(=010) = 2 (Operand)

NOT

- **Memory M Address :** 0000 0000 0000 1000
- **Binary code for bits 0 to 15 :** 0000 0100 011 010 000
- Reasons-
    a. **Opcode** : 0000 0100 as this is the address of NOT instruction in Control Memory.
    b. **8 to 6** : Store in Reg[3] as DR(=011) = 3
    c. **5 to 3** : Not Reg[2] as SA(=010) = 2
    d. **2 to 0.** : No Register B

ADD

- **Memory M Address :** 0000 0000 0000 1001
- **Binary code for bits 0 to 15 :** 0000 0101 101 010 010
- Reasons-
    a. **Opcode** : 0000 0101 as this is the address of NOT instruction in Control Memory.
    b. **8 to 6** : Store in Reg[5] as DR(=101) = 5
    c. **5 to 3** : Add Reg[2] as SA(=010) = 2
    d. **2 to 0.** : Add Reg[2] as SB(=010) = 2

ST

- **Memory M Address :** 0000 0000 0000 1010
- **Binary code for bits 0 to 15 :** 0000 0010 000 101 110
- Reasons-
    a. **Opcode** : 0000 0010 as this is the address of ST instruction in Control Memory.
    b. **8 to 6** : No Result (000)
    c. **5 to 3** : Store Reg[5] as SA(=101) = 5
    d. **2 to 0.** : Stored in Reg[6] as SB(=110) = 6

Question 2

Problems found in Q2:

Problem 1:

- SA never gets initialised when we start the process.
- In SRM1, we write R8 <- zf IR [2 : 0]. This loads the value of first 3 bits of Instruction Register and fill the rest with zeros.
- But, IR [2 : 0] is the same as SB and not SA, therefore SA is never initialised.
- To solve this problem we change SA to SB.

Problem 2:

- Changing SA to SB creates a different problem.
- At the end of the document, SB (previously SA) is assigned a value (DR = SB = 010).
- Along with that, IR[2 : 0] is assigned a value as well (zf IR[2:0] = 011).
- This contradicts the instructions as SB occupies the same space in machine code as IR[2 : 0] and both are assigned different values which is impossible.
- This problem can be solved if we add an instruction after SRM1 to load the correct SB value.

Problem 3:

- Problem arising due to addresses and conditions of SRM1, IF and SRM2.
- We load the Next Address of the instruction by using the conditions – by either incrementing the address or loading the address.
- The addresses are:
    - SRM1 is 0000 0111
    - SRM2 is 1000 0111
    - IF is 1100 0000 (which is taken from figure 5).
- When we look from SRM1, the Next Address can be IF or SRM2. If one out of the two (SRM2 or IF) is selected by loading the address, the other one has to be accessed by incrementing, which is not possible for any of the addresses.
- This problem can be solved if we add an instruction at 0000 1000 which would jump to the instruction which is unreachable (as explained in the previous point).