Saait Ahmed
ahmedsz

12/9/19
66192598

HW #13

1) i) 1: Fetch instruction 2: Decode instruction (Add mem <address> to Accumulator) 3: Fetch ADD operand, Fetch address, Fetch Accumulator 4: Execute, ADD address to Accumulator 5: Increment counter, return to step 2:

ii) 1: Fetch instruction 2: Decode (Store Accumulator to mem <address> 3: Fetch STORE operand, Fetch Accumulator, Fetch address. 4: Execute, Store the Accumulator in the mem <address>. 5: Increment counter, return to step 2:

2) It should have 32 bits. 14 of these bits would be operations on instructions. the 18 rest would be the Output ports. They would be weighted binary values. This computer could hold 2^{14} different instructions.

3) I will list what is missing from each section going top → down.

Instruction.	Machine Code
LDAA	C1
LDAA	0A
ADDA	C1
ADDA	0B
STAA	C1
HALT	3F

HW #13 Resub

1) A)

2: Instruction Fetch!
PC --> MAR
Memory Read
MBR -> IR

2: Decode
IF IR<Op Code> == ADD_FROM_MEMORY
THEN

3: Operand Gather
IR <addr> -> MAR
Memory Read

4. Instruction Execution
Memory -> ALU B
AC -> ALU A
ALU ADD
ALU Sum -> AC

5: Finish
PC+1 -> PC

1) B)

2: Instruction Fetch
PC -> MAR
MEMORY Read
MBR -> IR

2: Instruction Recode:
IF IR <Op Code> == STORE_TO_MEMORY
THEN

3: Operand Fetch:

~~IF IR <OP code> STORE~~

IF <addr> → MAR

4: Instruction Execution

AC → MBR

Memory Write

5: Housekeeping

PCH → PC

2) The number of bits should be the same as the memory width wise. The width is 18 bits so it should be 18 bits. It has 16K addresses which is 2^{14} meaning 14 bits were used. 4 bits are left for instructions which means 2^4 or 16 instructions

3) Missing machine code top to bottom

B6
B6
2A
BB
FF
00