

		MINCOM			
High-level Language	Assembly-level Language	Task	Micro-Programming Code	Control Points	Description
Fetch micro code to read in next assembly instruction					
I = I + J	MOVTA <234h>	Loading I	011 0 00 <address of FETCH>	C3	Fetch MOVTA 234
			100 1 0000 0000 0000	12,C2	initialise MAR with address of I bits 0-11 of I.R.
			100 0 0000 0100 0000	6,C2	move I to storage register
			100 0 0010 0010 0000	5, 9,C2	move I to accumulator, increment Program counter
			0 11 0 00 <address of FETCH>	C3	Fetch ADDA 654h
	ADDA 654h	Loading J	100 1 0000 0000 0000	12,C2	initialise MAR with address of I bits 0-11 of I.R.
			100 0 0000 0100 0000	6,C2	move J to storage register
		Add I to J	100 0 0000 0000 0110	1,2,C2	add storage register to accumulator in ALU
	MOVFA <234h>	Saving I	100 0 0010 0000 1000	3,9,C2	put result in accumulator, increment Program counter
			011 0 00 <address of FETCH>	C3	Fetch MOVFA <234h>
			100 1 0000 0000 0000	12,C2	initialise MAR with address of I bits 0-11 of I.R.
			100 0 0000 0000 0001	0,C2	move I to storage register from accumulator
	FETCH		100 0 0010 1000 0000	7,9,C2	move from storage register to RAM
			011 0 00 <address of FETCH>	C3	Fetch next instruction
			100 0 0100 0000 0000	10,C2	MAR <- PC
			100 0 0000 0100 0000	6,C2	fetch instruction to storage register
			100 0 0001 0000 0000	8,C2	storage register to instruction register
			000 0 0000 0000 0000	C0 , C1	branch to micro-code of instruction just fetched

Note that micro-code opcodes beginning with 1 XX cause the micro-program counter to be incremented by 1 as the control points are setup across C2
assume for demonstration purposes that the variable I is found in RAM address <234h>ex and J in 654hex
? does fetch increment PC or microcode of instruction

instruction mnemonic	opcode value	hex	
MOVTA <address>	0000	0	
MOVFA <address>	0001	1	
CLRA	0010	2	BASED ON DIAGRAM, ADDITION SHOULD BE ROUTED THROUGH ALU i.e. points 1,2,3
BNEG <address>	0011	3	
BRA <address>	0100	4	
ADDA <address>	0101	5	
MULT2 <address>	0110	6	
MOVIR	0111	7	
CLRML <address>	1000	8	
MULT4 <address>	1001	9	