BLG-222E Computer Organization Project-2 Demo Test Vectors

Initial Values

PC = 00 0000 0010 => 0x0002

Memory[002] = 0000 0000 0000 0100 => 0x0004

Memory[004] = 1010 1111 0000 0000 => 0xAF00

1. Instruction Fetch Test

Task: Fetch the instruction

RTL	**Required Control Signals**

Task Outcome:

2. Decode Test

Task: Decode the instruction

RTL	**Required Control Signals**

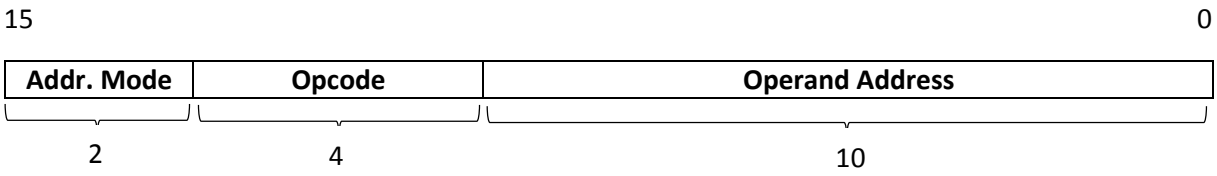
3. Operand Fetch and Execude Test

Assume that the operation is a direct logical or arithmetic operation and the effective address is already stored in AR.

The opcode from the previous steps is **0000**

Task Case-A: Assume the opcode 0000 is an Addition operation, perform operand fetch and execute	**Task Case-B:** Assume the opcode 0000 is an NOR operation, perform operand fetch and execute
RTL **Required Control Signals**	**RTL** **Required Control Signals**
Task Outcomes:	**Task Outcomes:**

Instruction Format



Addressing Information

Addressing Mode Bits	Effective Address	Mode
00	EA <- IR (9-0)	Direct
01	EA <- M [IR (9-0)]	Indirect
10	EA <- IR (9-0) + X	Indexed
11	EA <- SP	Stacked

ALU Operations Table

S_3	S_2	S_1	S_0	C (bit 0) in CCR	Operation	Function	Flag updates Z	N	O	C
1	0	0	0	0	$F \leftarrow A$	Transfer A	√	√	√	√
1	0	0	0	1	$F \leftarrow A + 1$	Increment A	√	√	√	√
1	0	0	1	0	$F \leftarrow A + B$	Addition	√	√	√	√
1	0	0	1	1	$F \leftarrow A + B + 1$	Add with carry	√	√	√	√
1	X	1	0	0	$F \leftarrow A + \bar{B}$	Subtract with borrow	√	√	√	√
1	0	1	0	1	$F \leftarrow A + \bar{B} + 1$	Subtraction	√	√	√	√
1	0	1	1	0	$F \leftarrow A - 1$	Decrement A	√	√	√	√
1	0	1	1	1	$F \leftarrow A$	Transfer A	√	√	√	√
0	1	0	0	0	$F \leftarrow A \wedge B$	AND	√	√	–	–
0	1	0	0	1	$F \leftarrow \overline{A \wedge B}$	NAND	√	√	–	–
0	1	1	0	0	$F \leftarrow A \vee B$	OR	√	√	–	–
0	1	1	0	1	$F \leftarrow A \oplus B$	XOR	√	√	–	–
0	1	0	1	0	$F \leftarrow \overline{A \vee B}$	NOR	√	√	–	–
0	1	0	1	1	$F \leftarrow \overline{A \oplus B}$	XNOR	√	√	–	–
0	1	1	1	X	$F \leftarrow \bar{A}$	Complement A	√	√	–	–
0	0	0	0	0	$F \leftarrow shrA$	Logical shift right A into F	√	√	–	–
0	0	0	0	1	$F \leftarrow ashrA$	Arithmetic shift right A into F	√	√	–	–
0	0	0	1	0	$F \leftarrow cshrA$	Circular shift right A into F	√	√	–	–
0	0	0	1	1	$F \leftarrow shlA$	Logical shift left A into F	√	√	–	–
0	0	1	X	0	$F \leftarrow ashlA$	Arithmetic shift left A into F	√	√	√	–
0	0	1	0	1	$F \leftarrow cshlA$	Circular shift left A into F	√	√	–	–