	Instruction	Operands	Description	Flags Affected	assembly code
1	ADD	REG, REG	operand1 = operand1 + operand2	ALL Flags	0-00000000 1-RRRRRR
2	AND	REG, REG	Logical AND between all bits of two operands. Result is stored in operand1.	OF=0 CF=0 SF,ZF are updated	0-00000001 0-RRRRRR
3	SUB	REG, REG	operand1 = operand1 - operand2 sub with 2's complement add	ALL Flags	0-00000001 1-RRRRRR
4	OR	REG, REG	Logical OR between all bits of two operands. Result is stored in operand1.	OF=0 CF=0 SF,ZF are updated	0-00000010 0-RRRRRR
5	XOR	REG, REG	Logical XOR between all bits of two operands. Result is stored in operand1.	OF=0 CF=0 SF,ZF are updated	0-00000010 1-RRRRRR
6	MOV	REG, REG	Copy operand2 to operand1	NONE	0-00000011 0-RRRRRR
7	ADC	REG	add operand with carry flag and stored result in operand	Same As ADD	0-00000011 1-RRRXXX
8	NOT	REG	One's complement negate operand1 = ~ operand1	NONE	0-00000100 0-RRRXXX
9	SAR	REG , immediate	Shift Arithmetic operand1 Right. The number of shifts is set by operand2. Shift all bits right, the bit that goes off is set to CF. The sign bit that is inserted to the left-most position has the same value as before shift.	OF=0 if first operand keeps original sign. the bit that goes off is set to CF. SF,ZF are updated after shifting	0-00000100 1- RRRIII
10	SLR	REG , immediate	Shift operand1 Right. (Logical) The number of shifts is set by operand2. Shift all bits right, the bit that goes off is set to CF. Zero bit is inserted to the left-most position.	OF=0 if first operand keeps original sign. the bit that goes off is set to CF. SF,ZF are updated after shifting	0-00000101 0- RRRIII
11	SAL	REG , immediate	Shift Arithmetic operand1 Left. The number of shifts is set by operand2.	OF=0 if first operand keeps original sign.	0-00000101 1- RRRIII

			Shift all bits left, the bit that goes off is set to CF. Zero bit is inserted to the right-most position.	the bit that goes off is set to CF. SF,ZF are updated after shifting	
12	SLL	REG , immediate	Shift operand1 Left. The number of shifts is set by operand2. Shift all bits left, the bit that goes off is set to CF. Zero bit is inserted to the right-most position.	OF=0 if first operand keeps original sign. the bit that goes off is set to CF. SF,ZF are updated after shifting	0-00000110 0- RRRIII
13	ROL	REG , immediate	Rotate operand1 right. The number of rotates is set by operand2. shift all bits right, the bit that goes off is set to CF and the same bit is inserted to the left-most position.	OF=0 if first operand keeps original sign. the bit that goes off is set to CF. SF,ZF are updated after shifting	0-00000110 1- RRRIII
14	ROR	REG , immediate	Rotate operand1 left. The number of rotates is set by operand2. shift all bits left, the bit that goes off is set to CF and the same bit is inserted to the right-most position.	OF=0 if first operand keeps original sign. the bit that goes off is set to CF. SF,ZF are updated after shifting	0-00000111 0- RRRIII
15	INC	REG	operand = operand + 1. Increment by 1	ALL (same as ADD)	0-00000111 1- RRRXXX
16	DEC	REG	operand = operand - 1. Decrement by 1	ALL (same as SUB)	0-00001000 0- RRRXXX
17	NOP	NONE	NONE	NONE	0-00000000 0- XXXXXX
18	ShowR	REG	Show Value of Register (operand) on LEDs	NONE	0-00001001 0- RRRXXX
19	ShowRR	REG, REG	concat the Value of two Register and show on LEDs	NONE	0-00001001 1- RRRRRR
20	LoadDipR	REG	Load 8 bits of Dip switches to the Register	NONE	0-00001010 0- RRRXXX
21	LoadDipRR	REG, REG	Load 16 bits of Dip switches to the two Registers	NONE	0-00001010 1-

					RRRRRR
22	CMP	REG, REG	operand1 - operand2 Result is not stored anywhere, flags are set (OF, SF, ZF, CF) according to result.	ALL	0-00001011 0-RRRRRR
23	JE	Address of Data Memory (IM)	address of memory that jumps to this if jump happened in operand2 and operand1 is don't care Condition: ZF = 1	NONE	1-0000-XXXAAAAAAAAA
24	JB	Address of Data Memory (IM)	Address of memory that jumps to this if jump happened in operand2 and operand1 is don't care. Condition: CF = 1 *Used for unsigned compare*	NONE	1-0001-XXXAAAAAAAAA
25	JA	Address of Data Memory (IM)	Address of memory that jumps to this if jump happened in operand2 and operand1 is don't care. Condition: CF = 0 , ZF=0 *Used for unsigned compare*	NONE	1-0010-XXXAAAAAAAAA
26	JL	Address of Data Memory (IM)	Address of memory that jumps to this if jump happened in operand2 and operand1 is don't care. Condition: SF not equal to OF *Used for signed compare*	NONE	1-0011-XXXAAAAAAAAA
27	JG	Address of Data Memory (IM)	address of memory that jumps to this if jump happened in operand2 and operand1 is don't care Condition: SF=OF , ZF=0 *Used for unsigned compare*	NONE	1-0100-XXXAAAAAAAAA
28	JMP	Address of Data Memory (IM)	Address of memory that jumps to this in operand2 and operand1 is don't care. Jump without any condition	NONE	1-0101-XXXAAAAAAAAA
29	LI	REG , immediate	Copy immediate value (operand2) to Register (operand1)	NONE	1-0110-RRRIIIIIIII
30	LM	REG, Address of Data Memory (DM)	Load data from data memory (address of memory in operand2) and copy to REG (operand1)	NONE	1-0111-RRRAAAAAAAAA
31	ShowDM	Address of Data Memory (DM)	Show data of data memory on LEDs (address of memory in operand2 and operand1 is don't care)	NONE	1-1000-XXXAAAAAAAAA
32	ShowIM	Address of Data Memory (IM)	Show data of instruction memory on LEDs (address of memory in operand2 and operand1 is don't care)	NONE	1-1001-XXXAAAAAAAAA