

## 006 AIR r immmed: Add immediate to Register

**CPU Simulator**

**Code Editor**

Line	Instruction	Binary Value
0	AIR 0,5	000110 0000000000100
1	AIR 1,0	000110 0100000000000
2	AIR 2,3	000110 1000000000011
3	AIR 3,10	000110 110000001010

**Action**

Reset

Run

Debug (Single Step)

Debug (Single Cycle)

run

**Keyboard (DEVID:0)**

Link to keyboard

1	2	3	-
4	5	6	*
7	8	9	/
0	.	Enter	

**Printer (DEVID:1)**

**Cache**

Enable Cache

HitRate: 80.00% (hit: 4, miss: 1)

Hover on a word to show value

Line	Value
Line 0	0001 1000 0000 0001 01
Line 1	10001 1001 0000 0000 00
Line 2	20001 1010 0000 0000 11
Line 3	30001 1011 0000 0010 10
Line 4	40000 0000 0000 0000 00
Line 5	50000 0000 0000 0000 00
Line 6	60000 0000 0000 0000 00
Line 7	70000 0000 0000 0000 00
Line 8	80000 0000 0000 0000 00
Line 9	90000 0000 0000 0000 00
Line 10	00000 0000 0000 0000 00
Line 11	00000 0000 0000 0000 00
Line 12	00000 0000 0000 0000 00
Line 13	00000 0000 0000 0000 00
Line 14	00000 0000 0000 0000 00
Line 15	00000 0000 0000 0000 00

**Console**

```

21:39:37[INFO]Reset registers,memory,cache
21:39:39[ERROR]line 5 cannot compile into binary code. Ignored
21:39:39[INFO]Load program into MEMORY
21:39:46[INSTR]PC = 0
21:39:46[CIRCLE]MAR = (PC)
21:39:46[CIRCLE]MBR = MEM(MAR)
21:39:46[CIRCLE]IR = (MBR)
21:39:46[CIRCLE]OPCODE (X R) ADDR = (R)
  
```

**Registers**

R1 (0) 000000000000100

**Memory**

Address	Binary Value
000110	0000000000100
000111	1000000000001

If immmed = 0, the Console shows ERROR, and the immmed cannot load in register (the R1 contains nothing)

## 007 SIR r, immmed: Subtract Immediate from Register

**CPU Simulator**

**Code Editor**

Line	Instruction	Binary Value
0	AIR 2,4	000110 1000000000100
1	SIR 2,1	000111 1000000000001

**Action**

Reset

Run

Debug (Single Step)

Debug (Single Cycle)

run

**Keyboard (DEVID:0)**

Link to keyboard

1	2	3	-
4	5	6	*
7	8	9	/
0	.	Enter	

**Printer (DEVID:1)**

**Cache**

Enable Cache

HitRate: N/A (hit: 0, miss: 0)

Hover on a word to show value

Line	Value
Line 0	0001 1010 0000 0001 00
Line 1	10001 1110 0000 0000 01
Line 2	20001 1100 0000 0000 01
Line 3	30000 0000 0000 0000 00
Line 4	40000 0000 0000 0000 00
Line 5	50000 0000 0000 0000 00
Line 6	60000 0000 0000 0000 00
Line 7	70000 0000 0000 0000 00
Line 8	80000 0000 0000 0000 00
Line 9	90000 0000 0000 0000 00
Line 10	00000 0000 0000 0000 00
Line 11	00000 0000 0000 0000 00
Line 12	00000 0000 0000 0000 00
Line 13	00000 0000 0000 0000 00
Line 14	00000 0000 0000 0000 00
Line 15	00000 0000 0000 0000 00

**Console**

```

15:25:46[CIRCLE]RES = OP1 - OP2
15:25:46[CIRCLE]R0 = (RES)
15:25:46[CIRCLE]PC = (PC) + 1
15:25:46[INSTR]PC = 3
15:25:46[CIRCLE]MAR = (PC)
15:25:46[CIRCLE]MBR = MEM(MAR)
15:25:46[CIRCLE]IR = (MBR)
15:25:46[INFO]Blank Instruction
  
```

**Registers**

CC 15 1111

**Memory**

Address	Binary Value
000110	0000000000100
000111	1000000000001

R<-c(r) - Immed

01 LDR r, x, address[,I]: Load register from memory.

**CPU Simulator**

**Code Editor**

Line	Instruction	Binary Value
0	AIR 0,1	000110 000000000001
1	AIR 2,5	000110 100000000101
2	STR 2,1,8	000010 100100001000
3	STR 0,1,5	000010 000100000101
4	LDR 1,2,8	000001 011000001000
5		

**Action**

Reset

Run (selected)

Debug (Single Step)

Debug (Single Cycle)

run

**Keyboard (DEVID:0)**

Link to keyboard

1	2	3	-
4	5	6	*
7	8	9	/
0	.	Enter	

**Console**

```
14:41:44[CIRCLE]MBR = MEM(MAR)
14:41:44[CIRCLE]R1 = (MBR)
14:41:44[CIRCLE]PC = (PC) + 1
14:41:44[INSTR]PC = 5
14:41:44[CIRCLE]MAR = (PC)
14:41:44[CIRCLE]MBR = MEM(MAR)
14:41:44[CIRCLE]IR = (MBR)
14:41:44[INFO]Blank Instruction
```

**File IO (DEVID:3,Type:INPUT)**

.\devid3.txt

Available Words:98 Change Reset

**File IO (DEVID:4,Type:OUTPUT)**

.\devid4.txt

Change Reset

**Printer (DEVID:1)**

Cache

Enable Cache

HitRate:77.78% (hit:7,miss:2)

Hover on a word to show value

Line	Value
Line 0	0000000000000000
Line 1	0000000000000000
Line 2	0000000000000000
Line 3	0000000000000000
Line 4	0000000000000000
Line 5	0000000000000000
Line 6	0000000000000000
Line 7	0000000000000000
Line 8	0000000000000000
Line 9	0000000000000000
Line 10	0000000000000000
Line 11	0000000000000000
Line 12	0000000000000000
Line 13	0000000000000000
Line 14	0000000000000000
Line 15	0000000000000000

**Registers**

<REG> Modify

Register	Value
R0 (1)	0
R1 (5)	5
R2 (5)	0
R3 (0)	0
X1 (0)	0
X2 (0)	0
X3 (0)	0
PC (5)	5
CC (0)	0
IR (1)	5
MAR (5)	5
MBR (1)	5
MSR (0)	0
MFR (0)	0
OPCODE (0)	0
R (1)	5
RX (0)	0
RY (0)	0
I (0)	0
X (2)	0
ADDR (8)	0
EA (8)	0
OP1 (0)	0
OP2 (5)	0
RES (5)	0
DEVID (0)	0

**Memory**

Address	Binary Value
0	0001 1000 0000 0000 01
1	0001 1010 0000 0001 01
2	0000 1010 0100 0010 00
3	0000 1000 0100 0001 01
4	0000 0101 1000 0010 00
5	0000 0000 0000 0000 01
6	0000 0000 0000 0000 00
7	0000 0000 0000 0000 00
8	0000 0000 0000 0001 01
9	0000 0000 0000 0000 00

Put the content (value 5) of the memory whose address is 8 into the register (R1). The offset in X2 is 0, so EA is 8. Therefore, the value 5 was put into the R1.

**CPU Simulator**

**Code Editor**

Line	Instruction	Binary Value
0	AIR 0,1	000110 000000000001
1	AIR 2,5	000110 100000000101
2	STR 2,1,8	000010 100100001000
3	STR 0,1,5	000010 000100000101
4	LDR 1,2,8	000001 011000001000
5		

**Action**

Reset

Run (selected)

Debug (Single Step)

Debug (Single Cycle)

run

**Keyboard (DEVID:0)**

Link to keyboard

1	2	3	-
4	5	6	*
7	8	9	/
0	.	Enter	

**Console**

```
14:42:19[CIRCLE]MBR = MEM(MAR)
14:42:19[CIRCLE]R1 = (MBR)
14:42:19[CIRCLE]PC = (PC) + 1
14:42:19[INSTR]PC = 5
14:42:19[CIRCLE]MAR = (PC)
14:42:19[CIRCLE]MBR = MEM(MAR)
14:42:19[CIRCLE]IR = (MBR)
14:42:19[INFO]Blank Instruction
```

**File IO (DEVID:3,Type:INPUT)**

.\devid3.txt

Available Words:98 Change Reset

**File IO (DEVID:4,Type:OUTPUT)**

.\devid4.txt

Change Reset

**Printer (DEVID:1)**

Cache

Enable Cache

HitRate:N/A (hit:0,miss:0)

L3W7:0000 0000 0000 0000 00

Line	Value
Line 0	0000000000000000
Line 1	0000000000000000
Line 2	0000000000000000
Line 3	0000000000000000
Line 4	0000000000000000
Line 5	0000000000000000
Line 6	0000000000000000
Line 7	0000000000000000
Line 8	0000000000000000
Line 9	0000000000000000
Line 10	0000000000000000
Line 11	0000000000000000
Line 12	0000000000000000
Line 13	0000000000000000
Line 14	0000000000000000
Line 15	0000000000000000

**Registers**

<REG> Modify

Register	Value
R0 (1)	0
R1 (1)	5
R2 (5)	0
R3 (0)	0
X1 (0)	0
X2 (0)	0
X3 (0)	0
PC (5)	5
CC (0)	0
IR (1)	5
MAR (5)	5
MBR (1)	5
MSR (0)	0
MFR (0)	0
OPCODE (0)	0
R (1)	5
RX (0)	0
RY (0)	0
I (1)	0
X (2)	0
ADDR (8)	0
EA (5)	0
OP1 (0)	0
OP2 (5)	0
RES (5)	0
DEVID (0)	0

**Memory**

Address	Binary Value
0	0001 1000 0000 0000 01
1	0001 1010 0000 0001 01
2	0000 1010 0100 0010 00
3	0000 1000 0100 0001 01
4	0000 0101 1010 0010 00
5	0000 0000 0000 0000 01
6	0000 0000 0000 0000 00
7	0000 0000 0000 0000 00
8	0000 0000 0000 0001 01
9	0000 0000 0000 0000 00

If we set I bit = 1, the value 5 would be the address and the value 1 was put into the R1.

**02 STR r, x, address[,I]: Store register to memory**

**41 LDX x, address[,I]: load index register from memory**

The screenshot shows a CPU Simulator interface with the following components:

- Code Editor:** Contains assembly instructions:
 

0	AIR 0,5	000110 000000000101
1	AIR 1,0	000110 010000000000
2	AIR 2,3	000110 100000000011
3	AIR 3,10	000110 110000001010
4	STR 2,1,7	000010 100100000111
5		
6		
- Action:** Includes buttons for Reset, Run (selected), Debug(SingleStep), Debug(SingleCycle), and a run button.
- Keyboard (DEVID:0):** A numeric keypad with buttons for digits 0-9, a decimal point, an equals sign, and an Enter key.
- Console:** Displays system logs:
 

```
23:13:45[CIRCLE]MBR = (R2)
23:13:46[CIRCLE]MEM(MAR) = MBR
23:13:46[CIRCLE]PC = (PC) + 1
23:13:46[INSTR]PC = 5
23:13:46[CIRCLE]MAR = (PC)
23:13:46[CIRCLE]MBR = MEM(MAR)
23:13:46[CIRCLE]IR = (MBR)
23:13:46[INFO]Blank Instruction
```
- Registers:** A table of 16 registers:
 

R0 (5)	R1 (0)	R2 (3)	R3 (10)
X1 (0)	X2 (0)	X3 (0)	PC (5)
CC (0)	IR (0)	MAR (5)	MBR (0)
MSR (0)	MFR (0)	OPCODE (0)	R (2)
RX (0)	RY (0)	I (0)	IX (1)
ADDR (7)	EA (7)	OP1 (0)	OP2 (10)
RES (10)	DEVID (0)		
- Memory:** A table showing memory addresses and binary values:
 

Address	Binary Value
2	0001 1010 0000 0000 11
3	0001 1011 0000 0010 10
4	0000 1010 0100 0001 11
5	0000 0000 0000 0000 00
6	0000 0000 0000 0000 00
7	0000 0000 0000 0000 11
8	0000 0000 0000 0000 00
9	0000 0000 0000 0000 00
10	0000 0000 0000 0000 00
11	0000 0000 0000 0000 00
- Cache:** Includes an 'Enable Cache' checkbox and a HitRate: N/A (hit:0, miss:0) indicator. Below is a 16x16 grid representing the cache lines (Line 0 to Line 15).

Put the content (value 3) in R2 into the memory whose address is 7 with the offset is 0 (in X1).

CPU Simulator

Code Editor

```

AIR 0,5      000110 000000000101
AIR 1,0      000110 010000000000
AIR 2,3      000110 100000000011
AIR 3,10     000110 1100000001010
STR 2,1,7    000010 100100000111
LDX 1,7      101001 000100000111
STR 2,1,8    000010 100100001000

```

Load

Action

Reset

☒ Run

☐ Debug(SingleStep)

☐ Debug(SingleCycle)

run

Keyboard (DEVID:0)

☐ Link to keyboard

1	2	3	-
4	5	6	*
7	8	9	/
0	.	Enter	

Console

```

23:02:44[CIRCLE]MBR = (R2)
23:02:44[CIRCLE]MEM(MAR) = MBR
23:02:44[CIRCLE]PC = (PC) + 1
23:02:44[INSTR]PC = 7
23:02:44[CIRCLE]MAR = (PC)
23:02:44[CIRCLE]MBR = MEM(MAR)
23:02:44[INFO]IR = (MBR)
23:02:44[INFO]Blank Instruction

```

☒ LOG ☒ ERROR ☒ CIRCLE ☒ INSTR ☒ WAIT

Registers

R1 4 0000000000000100 Modify

R0 (5)	R1 (0)	R2 (3)	R3 (10)
X1 (3)	X2 (0)	X3 (0)	PC (7)
CC (0)	IR (3)	MAR (7)	MBR (3)
MSR (0)	MFR (0)	OPCODE (0)	R (2)
RX (0)	RY (0)	I (0)	X (1)
ADDR (8)	EA (11)	OP1 (0)	OP2 (10)
RES (10)	DEVID (0)		

Memory

Address	Binary Value
2	0001 1010 0000 0000 11
3	0001 1011 0000 0010 10
4	0000 1010 0100 0001 11
5	1010 0100 0100 0001 11
6	0000 1010 0100 0010 00
7	0000 0000 0000 0000 11
8	0000 0000 0000 0000 00
9	0000 0000 0000 0000 00
10	0000 0000 0000 0000 00
11	0000 0000 0000 0000 11

Cache

☒ Enable Cache

HitRate: N/A (hit:0,miss:0)

Hover on a word to show value

Line 0	
Line 1	
Line 2	
Line 3	
Line 4	
Line 5	
Line 6	
Line 7	
Line 8	
Line 9	
Line 10	
Line 11	
Line 12	
Line 13	
Line 14	
Line 15	

Load index register (X1) from memory whose address is 7, so the X1 was loaded with 3. Put the content in R2 into the memory whose address is 8 with the offset is 3 (in X1), but the effect address (EA) is 11. So the memory whose address is 11 contains value 3.

**03 LDA r, x, address[,I]: Load register with address**

**CPU Simulator**

**Code Editor**

0	AIR 0,1	000110 000000000001
1	AIR 2,5	000110 100000000101
2	STR 2,1,9	000010 100100001001
3	LDX 2,9	101001 001000001001
4	STR 0,1,10	000010 000100001010
5	LDA 1,2,4	000011 011000000100

**Action**

Reset

☒ Run

☐ Debug(SingleStep)

☐ Debug(SingleCycle)

run

**Keyboard (DEVID:0)**

☐ Link to keyboard

1	2	3	-
4	5	6	*
7	8	9	/
0	.	Enter	

**Console**

```
00:26:59[CIRCLE]MAR = (ADDR,IX,I)
00:26:59[CIRCLE]R1 = (MAR)
00:26:59[CIRCLE]PC = (PC) + 1
00:26:59[INSTR]PC = 6
00:26:59[CIRCLE]MAR = (PC)
00:26:59[CIRCLE]MBR = MEM(MAR)
00:26:59[CIRCLE]IR = (MBR)
00:26:59[INFO]Blank Instruction
```

☒ LOG ☒ ERROR ☒ CIRCLE ☒ INSTR ☒ WAIT

**Registers**

R1 4 0000000000000100 Modify

R0 (1)	R1 (9)	R2 (5)	R3 (0)
X1 (0)	X2 (5)	X3 (0)	PC (6)
CC (0)	IR (0)	MAR (6)	MBR (0)
MSR (0)	MFR (0)	OPCODE (0)	R (1)
RX (0)	RY (0)	I (0)	IX (2)
ADDR (4)	EA (9)	OP1 (0)	OP2 (5)
RES (5)	DEVID (0)		

**Memory**

Address	Binary Value
2	0000 1010 0100 0010 01
3	1010 0100 1000 0010 01
4	0000 1000 0100 0010 10
5	0000 1101 1000 0001 00
6	0000 0000 0000 0000 00
7	0000 0000 0000 0000 00
8	0000 0000 0000 0000 00
9	0000 0000 0000 0001 01
10	0000 0000 0000 0000 01
11	0000 0000 0000 0000 00

**Cache**

☒ Enable Cache

HitRate:80.00% (hit:8,miss:2)

Hover on a word to show value

Line 0					
Line 1					
Line 2					
Line 3					
Line 4					
Line 5					
Line 6					
Line 7					
Line 8					
Line 9					
Line 10					
Line 11					
Line 12					
Line 13					
Line 14					
Line 15					

Put the content (value 5) of the memory whose address is 4 into the register (R1), but the offset in X2 is 5, so EA is 9. Therefore, the address 9 was put into the R1.

#### 42 STX x, address[,I]: store index register to memory

**CPU Simulator**

**Code Editor**

0	AIR 0,1	000110 000000000001
1	AIR 2,5	000110 100000000101
2	STR 2,1,9	000010 100100001001
3	LDX 2,9	101001 001000001001
4	STX 2,10	101010 001000001010

**Action**

Reset

☒ Run

☐ Debug(SingleStep)

☐ Debug(SingleCycle)

run

**Keyboard (DEVID:0)**

☐ Link to keyboard

1	2	3	-
4	5	6	*
7	8	9	/
0	.	Enter	

**Console**

```
00:30:50[CIRCLE]MBR = (X2)
00:30:50[CIRCLE]MEM(MAR) = MBR
00:30:50[CIRCLE]PC = (PC) + 1
00:30:50[INSTR]PC = 5
00:30:50[CIRCLE]MAR = (PC)
00:30:50[CIRCLE]MBR = MEM(MAR)
00:30:50[CIRCLE]IR = (MBR)
00:30:50[INFO]Blank Instruction
```

☒ LOG ☒ ERROR ☒ CIRCLE ☒ INSTR ☒ WAIT

**Registers**

R1 4 0000000000000100 Modify

R0 (1)	R1 (0)	R2 (5)	R3 (0)
X1 (0)	X2 (5)	X3 (0)	PC (5)
CC (0)	IR (0)	MAR (5)	MBR (0)
MSR (0)	MFR (0)	OPCODE (0)	R (0)
RX (0)	RY (0)	I (0)	IX (2)
ADDR (10)	EA (10)	OP1 (0)	OP2 (5)
RES (5)	DEVID (0)		

**Memory**

Address	Binary Value
2	0000 1010 0100 0010 01
3	1010 0100 1000 0010 01
4	1010 1000 1000 0010 10
5	0000 0000 0000 0000 00
6	0000 0000 0000 0000 00
7	0000 0000 0000 0000 00
8	0000 0000 0000 0000 00
9	0000 0000 0000 0001 01
10	0000 0000 0000 0001 01
11	0000 0000 0000 0000 00

**Cache**

☒ Enable Cache

HitRate:77.78% (hit:7,miss:2)

Hover on a word to show value

Line 0					
Line 1					
Line 2					
Line 3					
Line 4					
Line 5					
Line 6					
Line 7					
Line 8					
Line 9					
Line 10					
Line 11					
Line 12					
Line 13					
Line 14					
Line 15					

Put the content (value 5) in X2 into the memory whose address is 10, so the content in

memory whose address is 10 is value 5.

**010 JZ r, x, address[,I]: jump if zero**

**011 JNE r, x, address[,I]:jump if not equal**

CPU Simulator

Code Editor

```

0 AIR 0,1      000110 000000000001
1 AIR 2,5      000110 100000000101
2 STR 2,1,9    000010 100100001001
3 LDX 2,9      101001 001000001001
4 JZ 1,2,5     001010 011000000101
5

```

Load

Action

Reset

☒ Run

☐ Debug(Single Step)

☐ Debug(SingleCycle)

run

Keyboard (DEVID:0)

☐ Link to keyboard

1	2	3	-
4	5	6	*
7	8	9	/
0	.	Enter	

Console

```

00:47:05[CIRCLE]OPCODE.IX,R,I,ADDR = (IR)
00:47:05[CIRCLE]MAR = (ADDR,IX,I)
00:47:05[CIRCLE]PC = (MAR)
00:47:05[INSTR]PC = 10
00:47:05[CIRCLE]MAR = (PC)
00:47:05[CIRCLE]MBR = MEM(MAR)
00:47:05[CIRCLE]IR = (MBR)
00:47:05[INFO]Blank Instruction

```

☒ LOG ☒ ERROR ☒ CIRCLE ☒ INSTR ☒ WAIT

Registers

R1 4 0000000000000100 Modify

R0 (1)	R1 (0)	R2 (5)	R3 (0)
X1 (0)	X2 (5)	X3 (0)	PC (10)
CC (0)	IR (0)	MAR (10)	MBR (0)
MSR (0)	MFR (0)	OPCODE (0)	R (1)
RX (0)	RY (0)	I (0)	X (2)
ADDR (5)	EA (10)	OP1 (0)	OP2 (5)
RES (5)	DEVID (0)		

Memory

Address	Binary Value
2	0000 1010 0100 0010 01
3	1010 0100 1000 0010 01
4	0010 1001 1000 0001 01
5	0000 0000 0000 0000 00
6	0000 0000 0000 0000 00
7	0000 0000 0000 0000 00
8	0000 0000 0000 0000 00
9	0000 0000 0000 0001 01
10	0000 0000 0000 0000 00
11	0000 0000 0000 0000 00

Cache

☒ Enable Cache

HitRate:N/A (hit:0,miss:0)

L10WB:0000 0000 0000 0000 00

Line 0	
Line 1	
Line 2	
Line 3	
Line 4	
Line 5	
Line 6	
Line 7	
Line 8	
Line 9	
Line 10	
Line 11	
Line 12	
Line 13	
Line 14	
Line 15	

CPU Simulator

Code Editor

```

0 AIR 0,1      000110 000000000001
1 AIR 2,5      000110 100000000101
2 STR 2,1,9    000010 100100001001
3 LDX 2,9      101001 001000001001
4 JNE 0,2,5    001011 001000000101
5

```

Load

Action

Reset

☒ Run

☐ Debug(Single Step)

☐ Debug(SingleCycle)

run

Keyboard (DEVID:0)

☐ Link to keyboard

1	2	3	-
4	5	6	*
7	8	9	/
0	.	Enter	

Console

```

09:51:32[CIRCLE]OPCODE.IX,R,I,ADDR = (IR)
09:51:32[CIRCLE]MAR = (ADDR,IX,I)
09:51:32[CIRCLE]PC = (MAR)
09:51:32[INSTR]PC = 10
09:51:32[CIRCLE]MAR = (PC)
09:51:32[CIRCLE]MBR = MEM(MAR)
09:51:32[CIRCLE]IR = (MBR)
09:51:32[INFO]Blank Instruction

```

☒ LOG ☒ ERROR ☒ CIRCLE ☒ INSTR ☒ WAIT

Registers

R1 4 0000000000000100 Modify

R0 (1)	R1 (0)	R2 (5)	R3 (0)
X1 (0)	X2 (5)	X3 (0)	PC (10)
CC (0)	IR (0)	MAR (10)	MBR (0)
MSR (0)	MFR (0)	OPCODE (0)	R (0)
RX (0)	RY (0)	I (0)	X (2)
ADDR (5)	EA (10)	OP1 (0)	OP2 (5)
RES (5)	DEVID (0)		

Memory

Address	Binary Value
2	0000 1010 0100 0010 01
3	1010 0100 1000 0010 01
4	0010 1100 1000 0001 01
5	0000 0000 0000 0000 00
6	0000 0000 0000 0000 00
7	0000 0000 0000 0000 00
8	0000 0000 0000 0000 00
9	0000 0000 0000 0001 01
10	0000 0000 0000 0000 00
11	0000 0000 0000 0000 00

Cache

☒ Enable Cache

HitRate:75.00% (hit:0,miss:2)

Hover on a word to show value

Line 0	
Line 1	
Line 2	
Line 3	
Line 4	
Line 5	
Line 6	
Line 7	
Line 8	
Line 9	
Line 10	
Line 11	
Line 12	
Line 13	
Line 14	
Line 15	

For JZ, when  $c(R) = 0$  ( $R1 = 0$ ), the EA (value  $10 = 5 + 5$ ) was put into PC. Contrast

to JZ, for JNE when  $c(R) \neq 0$  ( $R0 = 1$ ), the EA (value 10 = 5 + 5) was put into PC.

**CPU Simulator**

**Code Editor**

Line	Instruction	Binary Value
0	AIR 0,1	000110 000000000001
1	AIR 2,5	000110 100000000101
2	STR 2,1,9	000010 100100001001
3	LDX 2,9	101001 001000001001
4	JZ 0,2,5	001010 001000000101

**Action**

Reset

Run

Debug(SingleStep)

Debug(SingleCycle)

run

**Keyboard (DEVID:0)**

Link to keyboard

1	2	3	-
4	5	6	*
7	8	9	/
0	.	Enter	

**Console**

```
00:58:00[CIRCLE]OPCODE.IX,R,I,ADDR = (IR)
00:58:00[CIRCLE]MAR = (ADDR,IX,I)
00:58:00[CIRCLE]PC = (MAR)
00:58:00[INSTR]PC = 5
00:58:00[CIRCLE]MAR = (PC)
00:58:00[CIRCLE]MBR = MEM(MAR)
00:58:00[CIRCLE]IR = (MBR)
00:58:00[INFO]Blank Instruction
```

**Registers**

R1 4 0000000000000100

Register	Value
R0 (1)	1
R1 (0)	0
R2 (5)	0
R3 (0)	0
X1 (0)	0
X2 (5)	0
X3 (0)	0
PC (5)	5
CC (0)	0
IR (0)	0
MAR (5)	5
MBR (0)	0
MSR (0)	0
MFR (0)	0
OPCODE (0)	0
R (0)	0
RX (0)	0
RY (0)	0
I (0)	0
IX (2)	0
ADDR (5)	5
EA (10)	10
OP1 (0)	0
OP2 (5)	5
RES (5)	5
DEVID (0)	0

**Memory**

Address	Binary Value
2	0000 1010 0100 0010 01
3	1010 0100 1000 0010 01
4	0010 1000 1000 0001 01
5	0000 0000 0000 0000 00
6	0000 0000 0000 0000 00
7	0000 0000 0000 0000 00
8	0000 0000 0000 0000 00
9	0000 0000 0000 0000 00
10	0000 0000 0000 0000 00
11	0000 0000 0000 0000 00

**Cache**

Enable Cache

HitRate: 75.00% (hit:6,miss:2)

L10W7:0000 0000 0000 0000 00

Line	Value
Line 0	0000 0000 0000 0000 00
Line 1	0000 0000 0000 0000 00
Line 2	0000 0000 0000 0000 00
Line 3	0000 0000 0000 0000 00
Line 4	0000 0000 0000 0000 00
Line 5	0000 0000 0000 0000 00
Line 6	0000 0000 0000 0000 00
Line 7	0000 0000 0000 0000 00
Line 8	0000 0000 0000 0000 00
Line 9	0000 0000 0000 0000 00
Line 10	0000 0000 0000 0000 00
Line 11	0000 0000 0000 0000 00
Line 12	0000 0000 0000 0000 00
Line 13	0000 0000 0000 0000 00
Line 14	0000 0000 0000 0000 00
Line 15	0000 0000 0000 0000 00

**CPU Simulator**

**Code Editor**

Line	Instruction	Binary Value
0	AIR 0,1	000110 000000000001
1	AIR 2,5	000110 100000000101
2	STR 2,1,9	000010 100100001001
3	LDX 2,9	101001 001000001001
4	JNE 1,2,5	001011 011000000101

**Action**

Reset

Run

Debug(SingleStep)

Debug(SingleCycle)

run

**Keyboard (DEVID:0)**

Link to keyboard

1	2	3	-
4	5	6	*
7	8	9	/
0	.	Enter	

**Console**

```
09:52:26[CIRCLE]OPCODE.IX,R,I,ADDR = (IR)
09:52:26[CIRCLE]MAR = (ADDR,IX,I)
09:52:26[CIRCLE]PC = (MAR)
09:52:26[INSTR]PC = 5
09:52:26[CIRCLE]MAR = (PC)
09:52:26[CIRCLE]MBR = MEM(MAR)
09:52:26[CIRCLE]IR = (MBR)
09:52:26[INFO]Blank Instruction
```

**Registers**

R1 4 0000000000000100

Register	Value
R0 (1)	1
R1 (0)	0
R2 (5)	0
R3 (0)	0
X1 (0)	0
X2 (5)	0
X3 (0)	0
PC (5)	5
CC (0)	0
IR (0)	0
MAR (5)	5
MBR (0)	0
MSR (0)	0
MFR (0)	0
OPCODE (0)	0
R (1)	1
RX (0)	0
RY (0)	0
I (0)	0
IX (2)	0
ADDR (5)	5
EA (10)	10
OP1 (0)	0
OP2 (5)	5
RES (5)	5
DEVID (0)	0

**Memory**

Address	Binary Value
2	0000 1010 0100 0010 01
3	1010 0100 1000 0010 01
4	0010 1101 1000 0001 01
5	0000 0000 0000 0000 00
6	0000 0000 0000 0000 00
7	0000 0000 0000 0000 00
8	0000 0000 0000 0000 00
9	0000 0000 0000 0000 00
10	0000 0000 0000 0000 00
11	0000 0000 0000 0000 00

**Cache**

Enable Cache

HitRate: 75.00% (hit:6,miss:2)

L8W7:0000 0000 0000 0000 00

Line	Value
Line 0	0000 0000 0000 0000 00
Line 1	0000 0000 0000 0000 00
Line 2	0000 0000 0000 0000 00
Line 3	0000 0000 0000 0000 00
Line 4	0000 0000 0000 0000 00
Line 5	0000 0000 0000 0000 00
Line 6	0000 0000 0000 0000 00
Line 7	0000 0000 0000 0000 00
Line 8	0000 0000 0000 0000 00
Line 9	0000 0000 0000 0000 00
Line 10	0000 0000 0000 0000 00
Line 11	0000 0000 0000 0000 00
Line 12	0000 0000 0000 0000 00
Line 13	0000 0000 0000 0000 00
Line 14	0000 0000 0000 0000 00
Line 15	0000 0000 0000 0000 00

For JZ, when  $c(R) \neq 0$  ( $R0 = 1$ ),  $PC = PC + 1$  (value 5 = 4 + 1). Contrast to JZ, for JNE when  $c(R) = 0$  ( $R1 = 0$ ),  $PC = PC + 1$  (value 5 = 4 + 1).

**013 JMP x, address[,I]: Unconditional Jump To Address**



**CPU Simulator**

**Code Editor**

0	AIR 0,1	000110 000000000001
1	AIR 2,5	000110 100000000101
2	STR 2,1,6	000010 100100000110
3	JMP 1,6	001101 000100000110
4		

Load

**Action**

Reset

☒ Run

☐ Debug(SingleStep)

☐ Debug(SingleCycle)

run

**Keyboard (DEVID:0)**

☐ Link to keyboard

1	2	3	-
4	5	6	*
7	8	9	/
0	.	Enter	

**Console**

```

14:46:52[CIRCLE]OPCODE.IX,R,I,ADDR = (IR)
14:46:52[CIRCLE]MAR = (ADDR,IX,I)
14:46:52[CIRCLE]PC = (MAR)
14:46:52[INSTR]PC = 6
14:46:52[CIRCLE]MAR = (PC)
14:46:52[CIRCLE]MBR = MEM(MAR)
14:46:52[CIRCLE]IR = (MBR)
14:46:52[INFO]Blank Instruction

```

☒ LOG ☒ ERROR ☒ CIRCLE ☒ INSTR ☒ WAIT

**File IO (DevID:3,Type:INPUT)**

.\devid3.txt

Available Words:98 Change Reset

**File IO (DevID:4,Type:OUTPUT)**

.\devid4.txt

Change Reset

**Printer (DEVID:1)**

**Cache**

☒ Enable Cache

HitRate:83.33% (hit:5,miss:1)

Hover on a word to show value

Line 0															
Line 1															
Line 2															
Line 3															
Line 4															
Line 5															
Line 6															
Line 7															
Line 8															
Line 9															
Line 10															
Line 11															
Line 12															
Line 13															
Line 14															
Line 15															

**Registers**

<REG>  Modify

R0 (1)	R1 (0)	R2 (5)	R3 (0)
X1 (0)	X2 (0)	X3 (0)	PC (6)
CC (0)	IR (5)	MAR (6)	MBR (5)
MSR (0)	MFR (0)	OPCODE (0)	R (0)
RX (0)	RY (0)	I (0)	IX (1)
ADDR (6)	EA (6)	OP1 (0)	OP2 (5)
RES (5)	DEVID (0)		

**Memory**

Address	Binary Value
0	0001 1000 0000 0000 01
1	0001 1010 0000 0001 01
2	0000 1010 0100 0001 10
3	0011 0100 0100 0001 10
4	0000 0000 0000 0000 00
5	0000 0000 0000 0000 00
6	0000 0000 0000 0001 01
7	0000 0000 0000 0000 00
8	0000 0000 0000 0000 00
9	0000 0000 0000 0000 00

PC <- EA (value 6), if I bit not set;

**CPU Simulator**

**Code Editor**

0	AIR 0,1	000110 000000000001
1	AIR 2,5	000110 100000000101
2	STR 2,1,6	000010 100100000110
3	JMP 1,6,1	001101 000110000110
4		

Load

**Action**

Reset

☒ Run

☐ Debug(SingleStep)

☐ Debug(SingleCycle)

run

**Keyboard (DEVID:0)**

☐ Link to keyboard

1	2	3	-
4	5	6	*
7	8	9	/
0	.	Enter	

**Console**

```

14:48:50[CIRCLE]OPCODE.IX,R,I,ADDR = (IR)
14:48:50[CIRCLE]MAR = (ADDR,IX,I)
14:48:50[CIRCLE]PC = (MAR)
14:48:50[INSTR]PC = 5
14:48:50[CIRCLE]MAR = (PC)
14:48:50[CIRCLE]MBR = MEM(MAR)
14:48:50[CIRCLE]IR = (MBR)
14:48:51[INFO]Blank Instruction

```

☒ LOG ☒ ERROR ☒ CIRCLE ☒ INSTR ☒ WAIT

**File IO (DevID:3,Type:INPUT)**

.\devid3.txt

Available Words:98 Change Reset

**File IO (DevID:4,Type:OUTPUT)**

.\devid4.txt

Change Reset

**Printer (DEVID:1)**

**Cache**

☒ Enable Cache

HitRate:85.71% (hit:6,miss:1)

Hover on a word to show value

Line 0															
Line 1															
Line 2															
Line 3															
Line 4															
Line 5															
Line 6															
Line 7															
Line 8															
Line 9															
Line 10															
Line 11															
Line 12															
Line 13															
Line 14															
Line 15															

**Registers**

<REG>  Modify

R0 (1)	R1 (0)	R2 (5)	R3 (0)
X1 (0)	X2 (0)	X3 (0)	PC (5)
CC (0)	IR (0)	MAR (5)	MBR (0)
MSR (0)	MFR (0)	OPCODE (0)	R (0)
RX (0)	RY (0)	I (1)	IX (1)
ADDR (6)	EA (5)	OP1 (0)	OP2 (5)
RES (5)	DEVID (0)		

**Memory**

Address	Binary Value
0	0001 1000 0000 0000 01
1	0001 1010 0000 0001 01
2	0000 1010 0100 0001 10
3	0011 0100 0110 0001 10
4	0000 0000 0000 0000 00
5	0000 0000 0000 0000 00
6	0000 0000 0000 0001 01
7	0000 0000 0000 0000 00
8	0000 0000 0000 0000 00
9	0000 0000 0000 0000 00

PC<- c(EA), if I bit set

**014 JSR x, address[,I]: Jump and Save Return Address**



**CPU Simulator**

**Code Editor**

0	AIR 0,1	000110 000000000001
1	AIR 2,5	000110 100000000101
2	STR 2,1,6	000010 100100000110
3	STR 0,1,3	000010 000100000011
4	JSR 1,6	001110 000100000110
5		

Load

**Action**

Reset

☒ Run

☐ Debug(SingleStep)

☐ Debug(SingleCycle)

run

**Keyboard (DEVID:0)**

☐ Link to keyboard

1	2	3	-
4	5	6	*
7	8	9	/
0	.	Enter	

**Console**

```

14:56:02[CIRCLE]PC = (PC) + 1
14:56:02[CIRCLE]R3 = (PC)
14:56:02[CIRCLE]PC = (MAR)
14:56:02[INSTR]PC = 6
14:56:02[CIRCLE]MAR = (PC)
14:56:02[CIRCLE]MBR = MEM(MAR)
14:56:02[CIRCLE]IR = (MBR)
14:56:02[INFO]Blank Instruction

```

☒ LOG ☒ ERROR ☒ CIRCLE ☒ INSTR ☒ WAIT

**File IO (DevID:3,Type:INPUT)**

.\devid3.txt

Available Words:98 Change Reset

**File IO (DevID:4,Type:OUTPUT)**

.\devid4.txt

Change Reset

**Printer (DEVID:1)**

**Registers**

<REG>  Modify

R0 (1)	R1 (0)	R2 (5)	R3 (5)
X1 (0)	X2 (0)	X3 (0)	PC (6)
CC (0)	IR (5)	MAR (6)	MBR (5)
MSR (0)	MFR (0)	OPCODE (0)	R (0)
RX (0)	RY (0)	I (0)	X (1)
ADDR (6)	EA (6)	OP1 (0)	OP2 (5)
RES (5)	DEVID (0)		

**Memory**

Address	Binary Value
0	0001 1000 0000 0000 01
1	0001 1010 0000 0001 01
2	0000 1010 0100 0001 10
3	0000 0000 0000 0000 01
4	0011 1000 0100 0001 10
5	0000 0000 0000 0000 00
6	0000 0000 0000 0001 01
7	0000 0000 0000 0000 00
8	0000 0000 0000 0000 00
9	0000 0000 0000 0000 00

**Cache**

☒ Enable Cache

HitRate:87.50% (hit:7,miss:1)

Hover on a word to show value

Line 0	Line 1	Line 2	Line 3	Line 4	Line 5	Line 6	Line 7	Line 8	Line 9	Line 10	Line 11	Line 12	Line 13	Line 14	Line 15

R3<-PC+1; PC<-EA

**CPU Simulator**

**Code Editor**

0	AIR 0,1	000110 000000000001
1	AIR 2,5	000110 100000000101
2	STR 2,1,6	000010 100100000110
3	STR 0,1,3	000010 000100000011
4	JSR 1,6,1	001110 000110000110
5		

Load

**Action**

Reset

☒ Run

☐ Debug(SingleStep)

☐ Debug(SingleCycle)

run

**Keyboard (DEVID:0)**

☐ Link to keyboard

1	2	3	-
4	5	6	*
7	8	9	/
0	.	Enter	

**Console**

```

14:57:41[CIRCLE]PC = (PC) + 1
14:57:41[CIRCLE]R3 = (PC)
14:57:41[CIRCLE]PC = (MAR)
14:57:41[INSTR]PC = 5
14:57:41[CIRCLE]MAR = (PC)
14:57:41[CIRCLE]MBR = MEM(MAR)
14:57:41[CIRCLE]IR = (MBR)
14:57:41[INFO]Blank Instruction

```

☒ LOG ☒ ERROR ☒ CIRCLE ☒ INSTR ☒ WAIT

**File IO (DevID:3,Type:INPUT)**

.\devid3.txt

Available Words:98 Change Reset

**File IO (DevID:4,Type:OUTPUT)**

.\devid4.txt

Change Reset

**Printer (DEVID:1)**

**Registers**

<REG>  Modify

R0 (1)	R1 (0)	R2 (5)	R3 (5)
X1 (0)	X2 (0)	X3 (0)	PC (5)
CC (0)	IR (0)	MAR (5)	MBR (0)
MSR (0)	MFR (0)	OPCODE (0)	R (0)
RX (0)	RY (0)	I (1)	X (1)
ADDR (6)	EA (5)	OP1 (0)	OP2 (5)
RES (5)	DEVID (0)		

**Memory**

Address	Binary Value
0	0001 1000 0000 0000 01
1	0001 1010 0000 0001 01
2	0000 1010 0100 0001 10
3	0000 0000 0000 0000 01
4	0011 1000 0110 0001 10
5	0000 0000 0000 0000 00
6	0000 0000 0000 0001 01
7	0000 0000 0000 0000 00
8	0000 0000 0000 0000 00
9	0000 0000 0000 0000 00

**Cache**

☒ Enable Cache

HitRate:88.89% (hit:8,miss:1)

Hover on a word to show value

Line 0	Line 1	Line 2	Line 3	Line 4	Line 5	Line 6	Line 7	Line 8	Line 9	Line 10	Line 11	Line 12	Line 13	Line 14	Line 15

PC <- c(EA), if I bit set.

**016 SOB r, x, address[,I]: Subtract One and Branch.**

**CPU Simulator**

**Code Editor**  

```

0 AIR 2,5      000110 100000000101
1 STR 2,1,8    000010 100100001000
2 LDX 2,8      101001 001000001000
3 SOB 2,2,1    010000 101000000001
4

```

**Action**  
Reset  
☒ Run  
☐ Debug(SingleStep)  
☐ Debug(SingleCycle)  
run

**Keyboard (DEVID:0)**  
☐ Link to keyboard  

1	2	3	-
4	5	6	*
7	8	9	/
0	.	Enter	

**Console**  

```

11:37:33[CIRCLE]R2 = R2 - 1
11:37:33[CIRCLE]MAR = (ADDR,IX,I)
11:37:33[CIRCLE]PC = (MAR)
11:37:33[INSTR]PC = 6
11:37:33[CIRCLE]MAR = (PC)
11:37:34[CIRCLE]MBR = MEM(MAR)
11:37:34[CIRCLE]IR = (MBR)
11:37:34[INFO]Blank Instruction

```

**File IO (DevID:3,Type:INPUT)**  
.devid3.txt  
Available Words:98 Change Reset  
**File IO (DevID:4,Type:OUTPUT)**  
.devid4.txt  
Change Reset

**Printer (DEVID:1)**

**Registers**  

<REG>				Modify
R0 (0)	R1 (0)	R2 (4)	R3 (0)	
X1 (0)	X2 (5)	X3 (0)	PC (6)	
CC (0)	IR (0)	MAR (6)	MBR (0)	
MSR (0)	MFR (0)	OPCODE (0)	R (2)	
RX (0)	RY (0)	I (0)	X (2)	
ADDR (1)	EA (6)	OP1 (0)	OP2 (5)	
RES (5)	DEVID (0)			

**Memory**  

Address	Binary Value
0	0001 1010 0000 0001 01
1	0000 1010 0100 0010 00
2	1010 0100 1000 0010 00
3	0100 0010 1000 0000 01
4	0000 0000 0000 0000 00
5	0000 0000 0000 0000 00
6	0000 0000 0000 0000 00
7	0000 0000 0000 0000 00
8	0000 0000 0000 0001 01
9	0000 0000 0000 0000 00

**Cache**  
☒ Enable Cache  
HitRate:71.43% (hit:5,miss:2)  
Have on a word to show value  

Line 0	0001 1010 0000 0001 01
Line 1	0000 1010 0100 0010 00
Line 2	1010 0100 1000 0010 00
Line 3	0100 0010 1000 0000 01
Line 4	0000 0000 0000 0000 00
Line 5	0000 0000 0000 0000 00
Line 6	0000 0000 0000 0000 00
Line 7	0000 0000 0000 0000 00
Line 8	0000 0000 0000 0000 00
Line 9	0000 0000 0000 0000 00
Line 10	0000 0000 0000 0000 00
Line 11	0000 0000 0000 0000 00
Line 12	0000 0000 0000 0000 00
Line 13	0000 0000 0000 0000 00
Line 14	0000 0000 0000 0000 00
Line 15	0000 0000 0000 0000 00

The value of R2 subtracts one and then returns the result to R2 (value 4 = 5 - 1). The value in R2 > 0 (value 4), so put the EA (value 6) into PC.

**CPU Simulator**

**Code Editor**  

```

0 AIR 2,5      000110 100000000101
1 STR 2,1,8    000010 100100001000
2 LDX 2,8      101001 001000001000
3 SOB 3,2,1    010000 111000000001
4

```

**Action**  
Reset  
☒ Run  
☐ Debug(SingleStep)  
☐ Debug(SingleCycle)  
run

**Keyboard (DEVID:0)**  
☐ Link to keyboard  

1	2	3	-
4	5	6	*
7	8	9	/
0	.	Enter	

**Console**  

```

11:36:46[CIRCLE]R3 = R3 - 1
11:36:46[CIRCLE]MAR = (ADDR,IX,I)
11:36:46[CIRCLE]PC = (MAR)
11:36:46[INSTR]PC = 4
11:36:46[CIRCLE]MAR = (PC)
11:36:46[CIRCLE]MBR = MEM(MAR)
11:36:46[CIRCLE]IR = (MBR)
11:36:46[INFO]Blank Instruction

```

**File IO (DevID:3,Type:INPUT)**  
.devid3.txt  
Available Words:98 Change Reset  
**File IO (DevID:4,Type:OUTPUT)**  
.devid4.txt  
Change Reset

**Printer (DEVID:1)**

**Registers**  

<REG>				Modify
R0 (0)	R1 (0)	R2 (5)	R3 (-1)	
X1 (0)	X2 (5)	X3 (0)	PC (4)	
CC (0)	IR (0)	MAR (4)	MBR (0)	
MSR (0)	MFR (0)	OPCODE (0)	R (3)	
RX (0)	RY (0)	I (0)	X (2)	
ADDR (1)	EA (6)	OP1 (0)	OP2 (5)	
RES (5)	DEVID (0)			

**Memory**  

Address	Binary Value
0	0001 1010 0000 0001 01
1	0000 1010 0100 0010 00
2	1010 0100 1000 0010 00
3	0100 0011 1000 0000 01
4	0000 0000 0000 0000 00
5	0000 0000 0000 0000 00
6	0000 0000 0000 0000 00
7	0000 0000 0000 0000 00
8	0000 0000 0000 0000 00
9	0000 0000 0000 0001 01

**Cache**  
☒ Enable Cache  
HitRate:71.43% (hit:5,miss:2)  
Have on a word to show value  

Line 0	0001 1010 0000 0001 01
Line 1	0000 1010 0100 0010 00
Line 2	1010 0100 1000 0010 00
Line 3	0100 0011 1000 0000 01
Line 4	0000 0000 0000 0000 00
Line 5	0000 0000 0000 0000 00
Line 6	0000 0000 0000 0000 00
Line 7	0000 0000 0000 0000 00
Line 8	0000 0000 0000 0000 00
Line 9	0000 0000 0000 0000 00
Line 10	0000 0000 0000 0000 00
Line 11	0000 0000 0000 0000 00
Line 12	0000 0000 0000 0000 00
Line 13	0000 0000 0000 0000 00
Line 14	0000 0000 0000 0000 00
Line 15	0000 0000 0000 0000 00

The value of R2 subtracts one and then returns the result to R3 (value -1 = 0 - 1). The value in R3 < 0 (value -1), so PC = PC + 1 (value 4).

## 004 AMR r, x, address[,I]: Add Memory To Register

**Code Editor**

0	AIR 2,4	000110 100000000100	Load
1	AIR 1,3	000110 010000000011	
2	STR 2,1,6	000010 100100000110	
3	AMR 1,2,6	000100 011000000110	
4			

**Action**

Reset

☒ Run

☐ Debug(Single Step)

☐ Debug(Single Cycle)

run

**Keyboard (DEVID:0)**

☐ Link to keyboard

1	2	3	-
4	5	6	*
7	8	9	/
0	.	Enter	

**Console**

```

14:29:52[CIRCLE]RES = OP1 + OP2
14:29:53[CIRCLE]R1 = (RES)
14:29:53[CIRCLE]PC = (PC) + 1
14:29:53[INSTR]PC = 4
14:29:53[CIRCLE]MAR = (PC)
14:29:53[CIRCLE]MBR = MEM(MAR)
14:29:53[CIRCLE]IR = (MBR)
14:29:53[INFO]Blank Instruction
  
```

**File IO (DEVID:3,Type:INPUT)**

.\devid3.txt

Available Words:98 Change Reset

**File IO (DEVID:4,Type:OUTPUT)**

.\devid4.txt

Change Reset

**Printer (DEVID:1)**

**Cache**

☒ Enable Cache

HitRate:85.71% (hit:8,miss:1)

Hover on a word to show value

Line 0															
Line 1															
Line 2															
Line 3															
Line 4															
Line 5															
Line 6															
Line 7															
Line 8															
Line 9															
Line 10															
Line 11															
Line 12															
Line 13															
Line 14															
Line 15															

**Registers**

<REG> Modify

R0 (0)	R1 (7)	R2 (4)	R3 (0)
X1 (0)	X2 (0)	X3 (0)	PC (4)
CC (0)	IR (0)	MAR (4)	MBR (0)
MSR (0)	MFR (0)	OPCODE (0)	R (1)
RX (0)	RY (0)	I (0)	X (2)
ADDR (6)	EA (6)	OP1 (3)	OP2 (4)
RES (7)	DEVID (0)		

**Memory**

Address	Binary Value
0	0001 1010 0000 0001 00
1	0001 1001 0000 0000 11
2	0000 1010 0100 0001 10
3	0001 0001 1000 0001 10
4	0000 0000 0000 0000 00
5	0000 0000 0000 0000 00
6	0000 0000 0000 0001 00
7	0000 0000 0000 0000 00
8	0000 0000 0000 0000 00
9	0000 0000 0000 0000 00

The value (value 4) of memory was added to the R1 (value 3), so the R1 store value 7 ( $7 = 4 + 3$ ).

## 005 SMR r, x, address[,I]: Subtract Memory From Register

**Code Editor**

0	AIR 2,4	000110 100000000100	Load
1	AIR 1,3	000110 010000000011	
2	STR 2,1,6	000010 100100000110	
3	STR 1,1,7	000010 010100000111	
4	SMR 2,1,7	000101 100100000111	
5			

**Action**

Reset

☒ Run

☐ Debug(Single Step)

☐ Debug(Single Cycle)

run

**Keyboard (DEVID:0)**

☐ Link to keyboard

1	2	3	-
4	5	6	*
7	8	9	/
0	.	Enter	

**Console**

```

15:03:41[CIRCLE]RES = OP1 - OP2
15:03:41[CIRCLE]R2 = (RES)
15:03:41[CIRCLE]PC = (PC) + 1
15:03:41[INSTR]PC = 5
15:03:41[CIRCLE]MAR = (PC)
15:03:41[CIRCLE]MBR = MEM(MAR)
15:03:41[CIRCLE]IR = (MBR)
15:03:41[INFO]Blank Instruction
  
```

**File IO (DEVID:3,Type:INPUT)**

.\devid3.txt

Available Words:98 Change Reset

**File IO (DEVID:4,Type:OUTPUT)**

.\devid4.txt

Change Reset

**Printer (DEVID:1)**

**Cache**

☒ Enable Cache

HitRate:88.89% (hit:8,miss:1)

Hover on a word to show value

Line 0															
Line 1															
Line 2															
Line 3															
Line 4															
Line 5															
Line 6															
Line 7															
Line 8															
Line 9															
Line 10															
Line 11															
Line 12															
Line 13															
Line 14															
Line 15															

**Registers**

<REG> Modify

R0 (0)	R1 (3)	R2 (1)	R3 (0)
X1 (0)	X2 (0)	X3 (0)	PC (5)
CC (0)	IR (0)	MAR (5)	MBR (0)
MSR (0)	MFR (0)	OPCODE (0)	R (2)
RX (0)	RY (0)	I (0)	X (1)
ADDR (7)	EA (7)	OP1 (4)	OP2 (3)
RES (1)	DEVID (0)		

**Memory**

Address	Binary Value
0	0001 1010 0000 0001 00
1	0001 1001 0000 0000 11
2	0000 1010 0100 0001 10
3	0000 1001 0100 0001 11
4	0001 0110 0100 0001 11
5	0000 0000 0000 0000 00
6	0000 0000 0000 0001 00
7	0000 0000 0000 0000 11
8	0000 0000 0000 0000 00
9	0000 0000 0000 0000 00

$R < -c(r) - c(EA).$

061 IN r, devid: Input Character To Register from Device

062 OUT r, devid: Output Character to Device from Register.

CPU Simulator

Code Editor

0IN 1, 0111101 000100000000Load

1OUT 1, 1111110 000100000001

2JMP 0, 0011101 000000000000

3

Action

Reset

Run

Debug(SingleStep)

Debug(SingleCycle)

Continue

Keyboard (DEVID:0)

Link to keyboard

+

123

456

789

0-Enter

Console

16:06:42[CIRCLE]OPCODE.IX,R,I,ADDR = (IR)  
16:06:42[CIRCLE]MAR = (ADDR,IX,I)  
16:06:42[CIRCLE]PC = (MAR)  
16:06:42[INSTR]PC = 0  
16:06:42[CIRCLE]MAR = (PC)  
16:06:42[CIRCLE]MBR = MEM(MAR)  
16:06:42[CIRCLE]IR = (MBR)  
16:06:42[CIRCLE]OPCODE.IX,R,I,ADDR = (IR)

LOG

ERROR

CIRCLE

INSTR

WAIT

File IO (DevID:3,Type:INPUT)

.\devid3.txt

Available Words:98ChangeReset

File IO (DevID:4,Type:OUTPUT)

.\devid4.txt

ChangeReset

Printer (DEVID:1)

123  
NHS

Cache

Enable Cache

HitRate:96.30% (hit:26,miss:1)  
Hove on a word to show value

Line 0

Line 1

Line 2

Line 3

Line 4

Line 5

Line 6

Line 7

Line 8

Line 9

Line 10

Line 11

Line 12

Line 13

Line 14

Line 15

Registers

CC151111Modify

R0(68)R1(83)R2(0)R3(0)

X1(0)X2(0)X3(0)PC(0)

CC(0)IR(250112)MAR(0)MBR(250112)

MSR(0)MFR(0)OPCODE(61)R(1)

RX(0)RY(0)I(0)X(0)

ADDR(0)EA(0)OP1(0)OP2(0)

RES(0)DEVID(0)

Memory

AddressBinary Value

01111 0100 0100 0000 00

11111 1000 0100 0000 01

20011 0100 0000 0000 00

30000 0000 0000 0000 00

40000 0000 0000 0000 00

50000 0000 0000 0000 00

60000 0000 0000 0000 00

70000 0000 0000 0000 00

80000 0000 0000 0000 00

90000 0000 0000 0000 00