The PIPPIN User's Guide

from MHuffman notes

PIPPIN Instruction Set

Ref: *The Analytical Engine* p. 211

Note:
The purpose and intent of the **AND** instruction is not clear and is not included in the following table. The **PIPPIN** simulator included on the CD and at the authors' Web site places 0 in the accumulator regardless of the input to **AND**.

Opcode | Operand		Assembly Instruction	Description
Binary	**Hex**		
00000000 bbbbbbbb	00 XX	ADD X	Add contents of referenced memory address to contents of accumulator. *Address Mode = Direct*
			Example: Add value stored at memory address 128 (10000000 binary, 80 hex, variable W) to contents of accumulator.
			00000000 10000000 | 00 10 | ADD W
00010000 bbbbbbbb	10 XX	ADD #*n*	Add immediate value to contents of accumulator. *Address Mode = Immediate*
			Example: Add the number 45 (00101101 binary, 2D hex) to accumulator:
			00010000 00101101 | 10 2D | ADD #45
00000001 bbbbbbbb	01 XX	SUB X	Subtract contents of referenced memory address from contents of accumulator. *Address Mode = Direct*
			Example: Subtract value stored at memory address 129 (10000001 binary, 81 hex, variable X) from accumulator.

			00000001 10000001	01 11	SUB X
00010001 bbbbbbbb	11 XX	SUB #*n*	Subtract immediate value from contents of accumulator. *Address Mode = Immediate*		
			Example: Subtract the number 27 (00011011 binary, 1B hex) from accumulator:		
			00010001 00011011	11 1B	SUB #27
00000010 bbbbbbbb	02 XX	MUL X	Multiply contents of accumulator by 8-bit value stored at referenced memory address. *Address Mode = Direct*		
			Example: Multiply accumulator by value stored at memory address 130 (10000010 binary, 82 hex, variable Y).		
			00000010 10000010	02 12	MUL Y
00010010 bbbbbbbb	12 XX	MUL #*n*	Multiply contnets of accumulator by immediate value. *Address Mode = Immediate*		
			Example: Multiply accumulator by the number 5 (00000101 binary, 05 hex):		
			00010010 00000101	12 05	MUL #5
00000011 bbbbbbbb	03 XX	DIV X	Divide contents of accumulator by 8-bit value stored at referenced memory address. *Address Mode = Direct*		
			Example: Divide accumulator by value stored at memory address 131 (10000011 binary, 83 Hex, variable Z).		
			00000011 10000011	03 13	DIV Z
00010011 bbbbbbbb	13 XX	DIV #*n*	Divide contnets of accumulator by immediate value. *Address Mode = Immediate*		
			Example: Divide accumulator by the number 10 (00001010 binary, 0A hex):		
			00010011 00001010	13 0A	DIV #10

00000100 bbbbbbbb	04 XX	LOD X	Load the accumulator with the 8-bit value stored at referenced memory address. *Address Mode = Direct*
			Example: Place value stored at memory address 132 (10000100, 84 Hex, variable T1) into accumulator.
		00000011 10000011	03 13	DIV Z
00010100 bbbbbbbb	13 XX	LOD #*n*	Load the accumulator with an immediated value. *Address Mode = Immediate*
			Example: Place the number 100 (01100100 binary, 64 hex) into accumulator.
		00010100 01100100	14 64	LOD #100
00000101 bbbbbbbb	05 XX	STO X	Store the contents of the accumulator into the referenced memory address. *Address Mode = Direct*
			Example: Save value of accumulator in memory address 133 (10000101 binary, 85 hex, variable T2).
		00000101 10000101	05 15	STO T2
00001001 00000000	09 00	NOT	If the accumulator contains 0 then set the accumulator to 1; otherwise set the accumulator to 1. Assume any non-zero value = TRUE and 0 = FALSE; NOT inverts the "truth" of the accumulator.
00001010 bbbbbbbb	0A XX	CPZ X	Compare X with Zero; if the contents of the referenced memory address = 0, set the accumulator to 1 (TRUE), otherwise set accumulator to 0 (FALSE). e.g. TRUE or FALSE: "Does X = 0? *Address Mode = Direct*
			Example: Set the accumulator to 1 (TRUE) or 0 (FALSE) depending on whether the value in memory address 129 (10000001 binary, 81 Hex, variable X) = 0.
		00001010 10000001	0A 11	CPZ X

<table>
<tr><td rowspan="3">00001011 bbbbbbbb</td><td rowspan="3">0B XX</td><td rowspan="3">CPL X</td><td colspan="3">Compare X with Zero; if the contents of the referenced memory address is LESS than 0, set the accumulator to 1 (TRUE), otherwise set accumulator to 0 (FALSE). e.g. TRUE or FALSE: "Is X < 0?
Address Mode = Direct</td></tr>
<tr><td colspan="3">Example: Set the accumulator to 1 (TRUE) or 0 (FALSE) depending on whether the value in memory address 130 (10000010 binary, 82 Hex, variable Y) < 0.</td></tr>
<tr><td>00001011 10000010</td><td>0B 10</td><td>CPL Y</td></tr>
<tr><td rowspan="3">00001100 bbbbbbbb</td><td rowspan="3">0C XX</td><td rowspan="3">JMP n</td><td colspan="3">Unconditional Jump: Set PC to n and execute instruction at that address.</td></tr>
<tr><td colspan="3">Example: Execute instruction at memory address 14 (00001110 binary, 0E hex).</td></tr>
<tr><td>00001110 00001110</td><td>0C 0E</td><td>JMP 14</td></tr>
<tr><td rowspan="3">00001101 bbbbbbbb</td><td rowspan="3">0D XX</td><td rowspan="3">JMZ n</td><td colspan="3">Conditional Jump: Set PC to n and execute instruction at that address IF the accumulator = 0; otherwise go to next instruction.</td></tr>
<tr><td colspan="3">Example: Execute instruction at memory address 4 (00000100 binary, 04 hex) IF the accumulator is 0.</td></tr>
<tr><td>00001101 00000100</td><td>0D 04</td><td>JMZ 4</td></tr>
<tr><td>00001110 00000000</td><td>0E 00</td><td>NOP</td><td colspan="3">No operation; do nothing and go to next instruction.</td></tr>
<tr><td>00001111 00000000</td><td>0F 00</td><td>HLT</td><td colspan="3">Halt execution; Control Unit does not fetch any more instructions.</td></tr>
</table>

[TOP]

PROGRAM 1:

Load 2 numbers and add them

Address	Instruction	Accum	W	X	Y
00	LOD #3	3	0	0	0
02	STO W	3	3	0	0

04	LOD #7	7	3	0	0
06	STO X	7	3	7	0
08	ADD W	10	3	7	0
10	STO Y	10	3	7	10
12	HLT	10	3	7	10
...	...				
128	W				
129	X				
130	Y				

PIPPIN memory after program execution:

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	14	03	05	80	14	07	05	81	00	80	05	82	0F	00	00	00
01	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..
07	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
08	03	07	0A	00	00	00	00	00	00	00	00	00	00	00	00	00
09	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..
0F	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Note: 2-byte instructions (opcode + operand) mapped at bottom of memory starting at address 00.
Note: Memory addresses 80 hex through 82 hex (128 - 130 decimal) represented by **W**, **X**, and **Y**

[TOP]

PIPPIN Simulator

PROGRAM 2:
Count down loop

Address	Instruction	Accum Pass 1	Accum Pass 2	Accum Pass 3
00	LOD X	3	2	1
02	SUB #1	2	1	0

04	JMZ 10	2	1	0
06	STO X	2	1	***
08	JMP 0	2	1	***
10	HLT	***	***	0
Initial Memory Values		**Memory Pass 1**	**Memory Pass 2**	**Memory Pass 3**
W (128)	???	???	???	???
X (129)	3	2	1	1
Y (130)	???	???	???	???

[TOP]

PIPPIN Simulator

Revised: 23 JUN 2004 11:09