16 BIT CPU
MADE BY JALIL AYMAN AFROZ

Logisim Version: Logisim evo 3.6.1.1 OS (Window/MAC/Linux: Windows 11

A. Please explain how many types of instructions are supported in your processor, and explain the format of each type of instructions (e.g., which bits are used as the operation or function code, which bits are used to index the 1st, 2nd or 3rd operand, and which bits are used to store the immediate number). You can draw figures to better explain your answer.

   All three types of instructions are supported namely, I-type, R-type, and J-type.

   For R-type instructions: X = 58583850 + 58641592 + 58729184 = 175954626
   Y = X MOD 7 = 175954626 MOD 7 = 1

   Therefore, Bits (1~3): write (destination) register Bits (4~6): read register 1 Bits (7~9): read register 2 Bits(10~13): opcode

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
(unused)		Op code bits				Source 2			Source 1			Destination			(unused)

I-type instructions: (The immediate values are passed as 6-bit signed representation of the integer number x)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Immediate bits		Op code bits				Source			Destination			Immediate bits			

J-type instructions:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Immediate bits		Op code bits				(Unused)						Immediate bits			

Note: orderof inserting the immediate bits (from left to right):
3,2,1,15,14,0
example,ifimmediate x = 9 = 001001

Then: 3rdbit= 0, 2nd bit = 0, 1st bit = 1, 15th bit = 0¸ 14th bit = 0¸ 0th bit = 1

B. Pleaseexplain theformat of each instruction (including the format of this instruction and its operation codes,and other information if needed).

li

li$r1, x

opcode = 0111

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Immediate x bits		0	1	1	1	(unused)			$r1			Immediate x bits			

add

add $r1, $r2, $r3

opcode = 1110

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
(unused)		1	1	1	0	$r2			$r3			$r1			(unused)

<table>
<tr><td>and</td><td>and $r1, $r2, $r3
opcode = 1100
<table><tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td colspan="2">(unused)</td><td>1</td><td>1</td><td>0</td><td>0</td><td colspan="3">$r2</td><td colspan="3">$r3</td><td colspan="3">$r1</td><td>(unused)</td></tr></table></td></tr>
<tr><td>or</td><td>or $r1, $r2, $r3
opcode = 1101
<table><tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td colspan="2">(unused)</td><td>1</td><td>1</td><td>0</td><td>1</td><td colspan="3">$r2</td><td colspan="3">$r3</td><td colspan="3">$r1</td><td>(unused)</td></tr></table></td></tr>
<tr><td>load</td><td>ld $r1, $r2
opcode = 0000
<table><tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td colspan="3">$r2</td><td colspan="3">$r1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table></td></tr>
<tr><td>store</td><td>st $r1, $r2
opcode = 0001
<table><tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td colspan="3">$r2</td><td colspan="3">$r1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table></td></tr>
<tr><td>move</td><td>move $r1, $2
opcode = 0110
<table><tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td colspan="3">$r2</td><td colspan="3">$r1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table></td></tr>
</table>

<table>
<tr><td></td><td></td></tr>
<tr><td>addi</td><td>addi $r1, $r2, x
opcode = 0110
<table><tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td colspan="2">Immediate
x bits</td><td>0</td><td>1</td><td>1</td><td>0</td><td colspan="3">$r2</td><td colspan="3">$r1</td><td colspan="4">Immediate
x bits</td></tr></table></td></tr>
<tr><td>andi</td><td>andi $r1, $r2, x
opcode = 0100
<table><tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td colspan="2">Immediate
x bits</td><td>0</td><td>1</td><td>0</td><td>0</td><td colspan="3">$r2</td><td colspan="3">$r1</td><td colspan="4">Immediate
x bits</td></tr></table></td></tr>
<tr><td>ori</td><td>ori $r1, $r2, x
opcode = 0101
<table><tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td colspan="2">Immediate
x bits</td><td>0</td><td>1</td><td>0</td><td>1</td><td colspan="3">$r2</td><td colspan="3">$r1</td><td colspan="4">Immediate
x bits</td></tr></table></td></tr>
<tr><td>ble</td><td>ble $r1, $r2, x
opcode = 0010
<table><tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td colspan="2">Immediate
x bits</td><td>0</td><td>0</td><td>1</td><td>0</td><td colspan="3">$r1</td><td colspan="3">$r2</td><td colspan="4">Immediate
x bits</td></tr></table></td></tr>
<tr><td>bne</td><td>bne $r1, $r2, x</td></tr>
</table>

<table>
<tr><td></td><td>Opcode = 0011
<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">Immediate x bits</td><td>0</td><td>0</td><td>1</td><td>1</td><td colspan="3">$r1</td><td colspan="3">$r2</td><td colspan="4">Immediate x bits</td></tr>
</table></td></tr>
<tr><td>jump</td><td>jump x
Opcode = 1000
<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">Immediate x bits</td><td>1</td><td>0</td><td>0</td><td>0</td><td colspan="6">(Unused)</td><td colspan="4">Immediate x bits</td></tr>
</table></td></tr>
<tr><td>call</td><td>call x
Opcode = 1010
<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">Immediate x bits</td><td>1</td><td>0</td><td>1</td><td>0</td><td colspan="6">(Unused)</td><td colspan="4">Immediate x bits</td></tr>
</table></td></tr>
<tr><td>rtn</td><td>rtn
Opcode = 1001
<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">Immediate x bits</td><td>1</td><td>0</td><td>0</td><td>1</td><td colspan="6">(Unused)</td><td colspan="4">Immediate x bits</td></tr>
</table></td></tr>
<tr><td>halt</td><td>halt</td></tr>
</table>

	Opcode = 1111

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
(unused)		1	1	1	1	(Unused)						(unused)			

C. Fill the following tables with the machine codes of each instruction of the testing programs:

Test program 1:

instruction	machine code (binary)	machine code (hex)
li $r1, 1	00 0111 000 000 0001	1C01
li $r2, 2	01 0111 000 001 0000	5C10
li $r3, 10	01 0111 000 010 0010	5C22
add $r2, $r1, $r2	00 1110 000 001 001 0	3812
ble $r2, $r3, -1	11 0010 001 010 1111	C8AF
halt	00 1111 0000000000	3C00

Test program 2:

instruction	machine code (binary)	machine code (hex)
li $r1, 6	11 0111 000 000 0000	DC00
li $r2, 5	10 0111 000 001 0001	9C11
andi $r3, $r1, 3	01 0100 000 010 0001	5021
ori $r4, $r3, 8	00 0101 010 011 0010	1532
halt	00 1111 0000000000	3C00

Test program 3:

instruction	machine code (binary)	machine code (hex)
li $r1, 6	11 0111 000 000 0000	dc00
li $r2, 5	10 0111 000 001 0001	9c11
and $r3, $r1, $r2	00 1100 000 001 010 0	3014
li $r8, 0	00 0111 0001110000	1c70
store $r3, $r8	0000011110100000	07a0
or $r4, $r1, $r2	0011010000010110	3416
li $r8, 1	0001110001110001	1c71
store $r4, $r8	0000011110100000	07a0
li $r8, 1	0001110001110001	1c71
load $r7, $r8	0000001111100000	03e0
halt	0011110000000000	3c00

Test program 4:

instruction	machine code (binary)	machine code (hex)
li $r1, 6	1101110000000000	DC00
li $r2, 4	10 0111 000 001 0000	9C10
call 7	11 1010 000000 0001	E801

move $r4, $r3 li	00 0110 010 011 0000	1930
$r1, 7 li $r2, 8	11 0111 000 000 0001	DC01
call 3 move $r5,	00 0111 000 001 0010	1C12
$r3	01 1010 000000 0001	6801
jump 3	00 0110 010 100 0000	1940
add $r3, $r1, $r2	01 1000 000000 0001	6001
rtn	00 1110 000 001 0100	3814
halt	00 1001 000000 0000	2400
	00 1111 000000 0000	3C00