ISA PROJECT REPORT
Hernandez

ISA Specifications

OPCODE	OPERAND
4 BITS	16 BITS

Total Fixed Instruction Width - 20 bits

Memory: (2^16 = 65536) 65K X 16

Binary twos complement arithmetic.

16 different registers.

R1	0000
R2	0001
R3	0010
R4	0011
R5	0100
R6	0101
R7	0110
R8	1000
R9	1001
R10	1010
R11	1011
R12	1100
R13	1101
R14	1110
R15	1111

Instruction Definitions

RESET: Will reset all registers to default values of 0. Since this instruction doesn’t modify any values it has no operands. Instead you must use padding to meet instruction width of 20 bits.

OPCODE	PADDING
4 BITS	16 BITS

COUT: Will display register value onto screen. It includes a 4 bit direct addressing register.

OPCODE	REGISTER	PADDING
4 BITS	4 BITS	12 BITS

CIN: Inserts an integer value into a register. Includes direct addressing 4 bit register.

OPCODE	REGISTER	PADDING
4 BITS	4 BITS	12 BITS

LOOP: Similar to C++ do while loop. Loops through set number of instructions. Continues to loop if register is greater than 0. Includes Immediate addressing decimal that is used to determine the number of previous instructions you want to loop through. Also includes a direct addressing register that is used to keep track of the remaining number of times to loop. For example if you want to loop through previous two instructions you would write LOOP 2 R3. In binary this would be 00110000000000100010

OPCODE	DECIMAL	REGISTER
4 BITS	12 BITS	4 BITS

SUM: Adds two different registers and stores result in a third register. This instruction has three direct addressing registers.

OPCODE	REGISTER	REGISTER	REGISTER	PADDING
4 BITS	4 BITS	4 BITS	4 BITS	4 BITS

SUB: Subtract two different registers and stores result in a third register. This instruction has three direct addressing registers.

OPCODE	REGISTER	REGISTER	REGISTER	PADDING
4 BITS	4 BITS	4 BITS	4 BITS	4 BITS

MULT: Multiplies two different registers and stores result in a third register. This instruction has three direct addressing registers.

OPCODE	REGISTER	REGISTER	REGISTER	PADDING
4 BITS	4 BITS	4 BITS	4 BITS	4 BITS

DIV: Divides two different registers and stores result in a third register. This instruction has three direct addressing registers.

OPCODE	REGISTER	REGISTER	REGISTER	PADDING
4 BITS	4 BITS	4 BITS	4 BITS	4 BITS

MOD: Finds remainder of two registers and stores result in third register. This instruction has three direct addressing registers.

OPCODE	REGISTER	REGISTER	REGISTER	PADDING
4 BITS	4 BITS	4 BITS	4 BITS	4 BITS

PUT: Puts decimal value into register. Includes immediate addressing decimal and a direct addressing register.

OPCODE	DECIMAL	REGISTER
4 BITS	12 BITS	4 BITS

RETURN: Stops the program. No operands.

OPCODE	PADDING
4 BITS	16 BITS

Instruction List

OPCODE	INSTRUCTION	DESCRIPTION	OPERAND	#
0000	RESET	Reset program to default values.	NONE	1
0001	COUT	Display value. Ex: COUT R1	4 bit register	2
0010	CIN	Insert value into register. EX: CIN R1	4 bit register	3
0011	LOOP	Loops through set number of instructions. Ex: LOOP 1 R5	12 bit decimal 4 bit register	4
0100	x	x	x	5
0101	x	x	x	6
0110	SUM	Adds two different registers and stores in a third register. Ex: SUM R1 R2 R3	Three 4 bit registers.	7
0111	SUB	Subtract two different registers and stores in third register. Ex: SUB R1 R2 R3.	Three 4 bit registers.	8
1000	MULT	Multiplies two different registers and stores in third register. Ex: MULT R1 R2 R3.	Three 4 bit registers.	9
1001	DIV	Divide two different registers and stores in third register. Ex: DIV R1 R2 R3.	Three 4 bit registers.	10
1010	MOD	Finds remainder of two registers and stores result in third register Ex: MOD R1 R2 R3	Three 4 bit registers.	11
1011	x	x	x	12

1100	x	x	x	13
1101	x	x	x	14
1110	PUT	Put decimal into register.	12 bit decimal 4 bit register	15
1111	RETURN	Stops the program.	NONE	16

Note: x reserved for future instructions.

Difference between this ISA and MARIE is this ISA uses 20 bits vs MARIE's 16 bits. It allows for a larger operand. The project ISA includes more arithmetic operations such as multiplication and division. It also includes a LOOP instruction. Both ISAs use 4 bit opcodes so the number of possible instructions stays the same. MARIE currently uses 13 instructions. The project ISA has 11 instructions.

Benchmark Mnemonic Code

Find running sum of x numbers.

Binary

```
00000000000000000000 - RESET
11100000000000010011 - PUT 1 R4
00100010000000000000 - CIN R3
00100001000000000000 - CIN R2
01100000000100000000 - SUM R1 R2 R1
01110010001100100000 - SUB R3 R4 R3
00110000000000110010 - LOOP 3 R3
00010000000000000000 - COUT R1
11110000000000000000 - RETURN
```