MODULE 5: Assembly Language + Processor Control + Examples

Lecture 5.2
MARIE Instruction Set

Prepared By:
- Scott F. Midkiff, PhD
- Luiz A. DaSilva, PhD
- Kendall E. Giles, PhD

Electrical and Computer Engineering
Virginia Tech

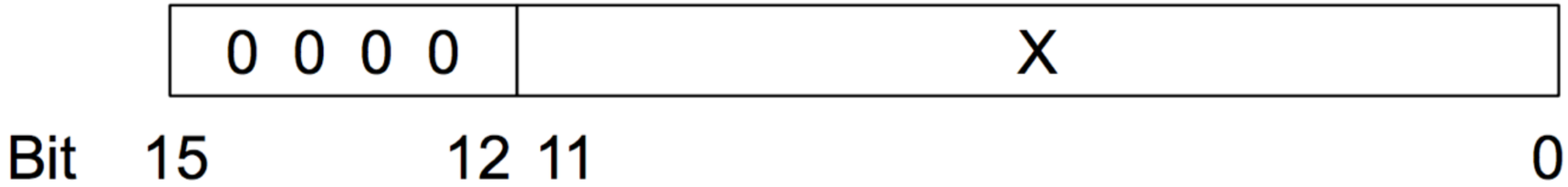
Lecture 5.2 Objectives

- Describe the function of by MARIE instructions JnS, Clear, AddI, and JumpI, and use these instructions in a program
- Distinguish between direct and indirect addressing
- Write MARIE programs that perform loops, subroutine calls, and if-then-else constructs

Additional Instructions

- We extend the previously discussed MARIE instruction set with four additional instructions
  - Jump and store (JnS)
  - Add indirect (AddI)
  - Jump indirect (JumpI)
  - Clear (Clear)

JnS X




```
MBR ← PC
MAR ← X
M[MAR] ← MBR
MBR ← X
AC ← 1
AC ← AC + MBR
PC ← AC
```

- Store a pointer to a return instruction
- Set the PC to a different instruction
- Enables calls to subroutines, then return to the calling point once the subroutine is finished

Clear

AC ← 0

- Moves all 0s into the accumulator
- Saves machine cycles that would be needed for loading a 0 from memory

Direct versus Indirect Addressing

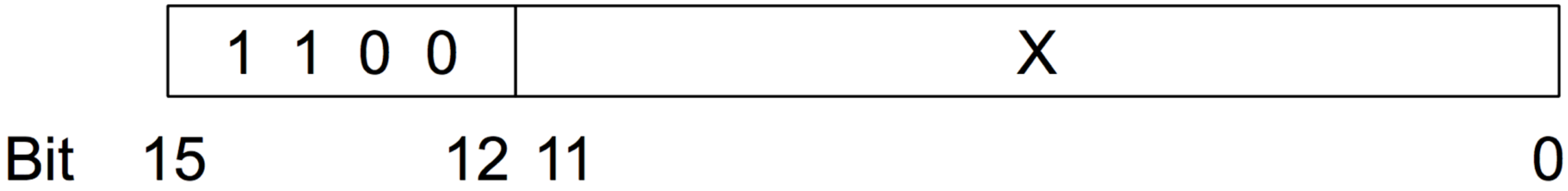
- Direct addressing: address of the operand is explicitly stated in the instruction
- So far, all of the MARIE instructions that we have discussed use a direct addressing mode
- Indirect addressing: the address of the address of the operand is given in the instruction
- If you have ever used pointers in a program, you are already familiar with indirect addressing

AddI X

MAR ← X
MBR ← M[MAR]
MAR ← MBR
MBR ← M[MAR]
AC ← AC + MBR

- Go to address X
- Use the value stored at location X as the address of the data operand to add to the AC

JumpI X



MAR ← X
MBR ← M[MAR]
PC ← MBR

- Go to address X
- Use the value stored at location X as the address of the location to jump to

CHECK POINT

As a checkpoint of your understanding, please pause the video and make sure you can do the following:

- Work through the RTL microoperations for the Jump and store, Add Indirect, Jump Indirect, and Clear instructions

If you have any difficulties, please review the lecture video before continuing.

Example: If-then-else Construct (1)

- Let us look at a program to perform the following:

```
If X = Y then
    X = X x 2
Else
    Y = Y - X
```

Example: If-then-else Construct (2)

```
100   If,    Load X
101          Subt Y          /AC = X - Y
102          Skipcond 400    /If AC=0, skip next instr.
103          Jump Else       /Jump to Else if AC != 0
104   Then,  Load X
105          Add X           /AC = 2 x X
106          Store X
107          Jump Endif
108   Else,  Load Y
109          Subt X          /AC = Y - X
10A          Store Y
10B   Endif, Halt
10C   X,     DEC 12
10D   Y,     DEC 10
```

Example: Loop to Add 5 Numbers

```
100         Load Addr       /AC = Address of 1st number
101         Store Next      /Use Next as a pointer
102         Load Num        /AC = # of items to be added
103         Subt One        /AC = AC - 1
104         Store Ctr       /Use Ctr to control the loop
105   Loop, Load Sum
106         AddI Next       /Add value pointed to by
                            /location Next
107         Store Sum
108         Load Next
109         Add One         /AC = AC + 1
10A         Store Next
10B         Load Ctr
10C         Subt One
10D         Store Ctr
10E         Skipcond 000    /If Ctr < 0, skip next instr.
10F         Jump Loop       /Otherwise, go back to Loop
110         Halt
111   Addr, HEX 117         /Numbers to be added start at
                            /location 118
112   Next, HEX 0           /Pointer to the next # to add
                            /location Next
113   Num,  DEC 5           /# of items to be added
114   Sum,  DEC 0           /The sum
115   Ctr,  HEX 0           /Control loop variable
116   One,  DEC 1           /Stores the integer 1
117         DEC 10          /Values to be added
(…)
```

CHECK POINT

As a checkpoint of your understanding, please pause the video and make sure you can do the following:

- Work through by hand the assembly instructions in the loop example on the previous slide

If you have any difficulties, please review the lecture video before continuing.

Summary

- The JnS instruction enables calls to subroutines, then a return to the calling point once the subroutine is finished
- The Clear instruction clears the AC
- In indirect addressing, the address of the address of the operand is included in the instruction
    - AddI and JumpI are instructions that use this mode of addressing
- While and for loops and if-then-else constructs can be realized with the limited MARIE instruction set

Lecture 5.2
MARIE Instruction Set

Prepared By:
- Scott F. Midkiff, PhD
- Luiz A. DaSilva, PhD
- Kendall E. Giles, PhD

Electrical and Computer Engineering
Virginia Tech