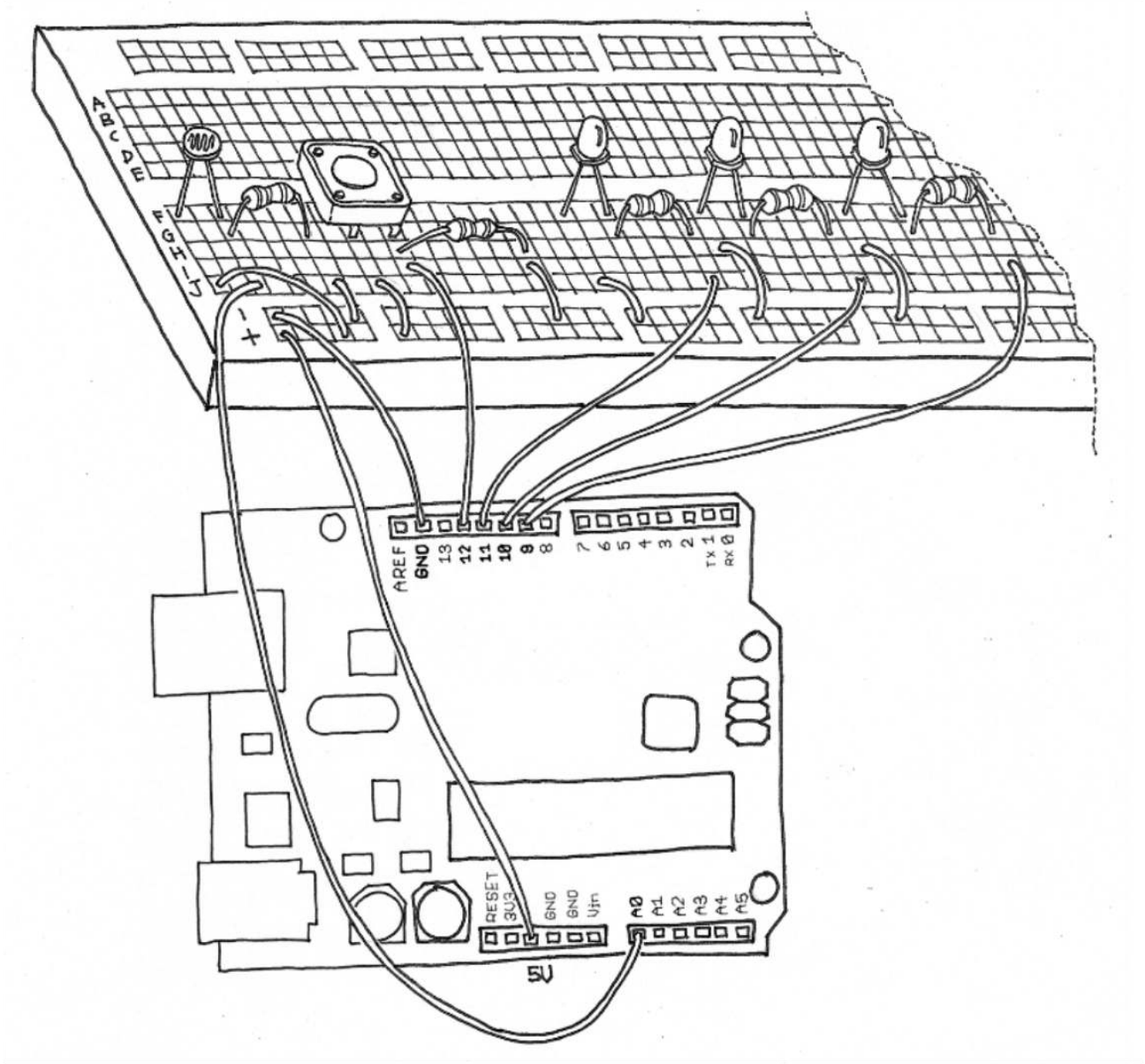
CS/EEE/INSTR F241
Microprocessor Programming and Interfacing

Lab 3 - Control Flow in ALP



Dr. Vinay Chamola and Anubhav Elhence

Introduction to Control Flow in ALP

What is CMP instruction?

The CMP instruction in Assembly language is part of the 80x86 instruction set and is used for comparing two values.

The CMP instruction compares two operands and sets the status flags in the flag register based on the result of the comparison. The operands can be immediate values, registers, or memory locations. The flags affected by the CMP instruction are the zero flag (ZF), carry flag (CF), sign flag (SF), overflow flag (OF), and auxiliary carry flag (AF).

Here's an example of how CMP can be used:

```
MOV AX, 5
CMP AX, 10 ; Compare AX (5) to 10
JAE LABEL ; Jump to LABEL if AX >= 10 (CF = 0)
; [.. do something if AX < 10 ..]
LABEL:
```

In this example, the CMP instruction compares the value in AX (5) with the value 10. If AX is greater than or equal to 10, the carry flag (CF) is cleared and the program will jump to the label LABEL. If AX is less than 10, the carry flag is set and the program will continue to the next instruction.

What is LODSB, LODSW, LODSD instruction?

The LODSB, LODSW, and LODSD instructions in Assembly language are part of the 80x86 instruction set and are used for loading data from memory into a register.

- **LODSB loads a byte (8-bit) value from the memory location pointed to by the source index (SI) into the destination register (AL). After the instruction is executed, the source index (SI) is incremented by 1.**
- **LODSW loads a word (16-bit) value from the memory location pointed to by the source index (SI) into the destination register (AX). After the instruction is executed, the source index (SI) is incremented by 2.**
- **LODSD loads a doubleword (32-bit) value from the memory location pointed to by the source index (SI) into the destination register (EAX). After the instruction is executed, the source index (SI) is incremented by 4.**

Here's an example of how LODSB can be used:

```
MOV SI, 1000H ; Load the starting address of the data into SI
MOV AL, 0 ; Clear AL

LOOP:
  LODSB ; Load the byte from memory into AL
  ; [.. do something with the data in AL ..]
  DEC CX ; Decrement the loop counter
  JNZ LOOP ; Repeat the loop if CX is not zero
```

In this example, **LODSB** loads the next byte from memory into AL, and the loop counter **CX** is decremented after each iteration. The loop continues as long as **CX** is not zero. The other instructions **LODSW** and **LODSD** work similarly, but load different sizes of data into the destination register.

What are the different variants of Jump Instruction in 8086 ?

The 80x86 processors have several variants of the Jump instruction. The most common variants are:

1. JMP: Unconditional jump, transfers control to a specified label or memory location.
2. JZ or JE: Jump if zero, transfers control to a specified label or memory location if the zero flag (ZF) is set.
3. JNZ or JNE: Jump if not zero, transfers control to a specified label or memory location if the zero flag (ZF) is not set.
4. JS: Jump if sign, transfers control to a specified label or memory location if the sign flag (SF) is set.
5. JNS: Jump if not sign, transfers control to a specified label or memory location if the sign flag (SF) is not set.
6. JO: Jump if overflow, transfers control to a specified label or memory location if the overflow flag (OF) is set.
7. JNO: Jump if not overflow, transfers control to a specified label or memory location if the overflow flag (OF) is not set.
8. JA or JNBE: Jump if above, transfers control to a specified label or memory location if the carry flag (CF) is not set and the zero flag (ZF) is not set.
9. JAE or JNB: Jump if above or equal, transfers control to a specified label or memory location if the carry flag (CF) is not set.
10. JB or JNAE: Jump if below, transfers control to a specified label or memory location if the carry flag (CF) is set.
11. JBE or JNA: Jump if below or equal, transfers control to a specified label or memory location if the carry flag (CF) is set or the zero flag (ZF) is set.

Here's an example of how JZ can be used:

```
MOV AX, 5
CMP AX, 10 ; Compare AX (5) to 10
JZ LABEL ; Jump to LABEL if AX = 10 (ZF = 1)
; [.. do something if AX ≠ 10 ..]
LABEL:
```

In this example, the CMP instruction compares the value in AX (5) with the value 10. If AX is equal to 10, the zero flag (ZF) is set and the program will jump to the label LABEL. If AX is not equal to 10, the zero flag is cleared and the program will continue to the next instruction.

Lab Task:

Task 1

Write an ALP that will examine the contents of set of 10 bytes starting from location 'array1' for the presence of data '0Ah' and replace it with ASCII character 'E'.

```
1   .model tiny
2   .data
3   array1        db      91h,02h,083h,0Ah,075h,0Ah,047h,012h,076h,61h
```

Task 2

Write an ALP that will count the number of negative numbers in an array of 16-bit signed data stored from location 'array1'. The number of elements in array1 is present in location 'count'. The count of negative numbers must be stored in location 'NEG1'

```
1   .model tiny
2   .data
3   array1        db      ;fill this up yourself
4   count         db      ;since you have filled the above array,
5                         ;therefore you only know the count
6   NEG1          db      ?
```


BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
ज्ञानं परमं बलम्