

EER-018
INTRODUCTION to DIGITAL COMPUTERS
LABORATORY NO. 8
SIMULATION of MICROCONTROLLER with SIMPLE I/O

OBJECTIVES:

1. Learn M68HC05 instruction subset.
2. Learn I/O registers.
3. Learn to assemble and run a program with input and output data.

EQUIPMENT :

Gateway2000, RAPID Microcontroller Development System, M68HC705JICS kit.

REFERENCES:

1. M68705J1A In-Circuit Simulator. User's Manual (copy in lab)
2. MC68HC705J1A Technical Data (you should have)

M68HC05 PROGRAM EXAMPLES:

Example 1.

In the following program we use only bit 7 of port A as an output. This program sets the output (put 1 at bit 7 of port A), then after a delay clears the output (put 0 at bit 7 of port A), and after a second delay, repeats. The set and reset is repeated 15 times. The following pseudo-code describes the behavior of the program.

```
set DDRA to $80
for (i=$F downto 0) do
    set bit 7 of PortA to 1;
    delay();
    set bit 7 of PortA to 0;
    delay();
end

delay():
    for (temp2 = $1F downto 0) do
        {
            for (temp3=$1F downto 0) do
                {nothing}
            }
        }
    }
```

The following program is the M68HC05 program ``Flashing LED''.

```
temp1    equ $C0                ;one byte temp storage location ; to store number of repetitions
temp2    equ $C1                ;one byte temp storage location ;for outer loop of the delay
temp3    equ $C2                ;one byte temp storage location ;for inner loop of the delay
org rom                ;program will start at $0300

start:    lda #$80                ;loads the hexadecimal number 80
          ;(binary 10000000) into the accumulator ACCA
          sta porta                ;stores the contents of ACCA ;(binary 10000000) in port A
          sta ddra                ;stores the contents of ACCA (binary 10000000)
          ;in data direction register bit 7 is to set to be an output
          lda #$F                ;loads the hexadecimal number F (decimal 15) into
          ;the accumulator ACCA
          sta temp1                ;stores the contents of ACCA (the hexadecimal number F
          ;(decimal 15)) in temp1 (the address $C0 )

loop0:    bset 7,porta            ;set PA7 and LED1 to 1
          bsr delay                ;branch to delay subroutine
          bclr 7,porta            ;clear PA7 and LED1 to 0
          bsr delay                ;branch to delay subroutine
          dec temp1                ; decrement repetition counter
          bne loop0                ;if counter is not =0, repeat loop0
```

```

        stop                ;stop oscillator of the microcontroller

delay:  lda #$1F            ;load ACCA with outer loop counter
        sta temp2          ;store in RAM in temp2
loop1:  lda #$1F            ;load ACCA with inner loop counter
        sta temp3          ;store in RAM in temp3
loop2:  dec temp3           ;decrement inner loop counter
        bne loop2          ;branch to inner loop if not zero
        dec temp2          ;decrement outer loop counter
        bne loop1          ;branch to outer loop if not zero
        rts                ;return from subroutine

        org $7FE           ;sets the address $07FE for the object code that follows
back1:  fdb start           ;forms double byte address constant start
        ;that defines where the program counter begins on reset

```

Example 2.

In the following program we use bit 7 of the port A as an output and bit 0 of the port A as an input. We can change the input using the switch S4. When S4 is pressed, a logic 1 is read on PA0. Otherwise a logic 0 is read. This program reads the PA0 input and places the result in the output an infinite number of times.

```

temp1   equ $C0            ;one byte temp storage location
        org rom           ;program will start at $0300
start:  lda #$80            ;loads the hexadecimal number 80 (decimal 128) into ACCA
        sta ddra           ;stores ACCA (the hexadecimal number 80 (binary 10000000))
        ;in data direction register - bit 7 is an output bit 0 is the input
        lda #$1            ;loads the hexadecimal number 1 (binary 00000001) into ACCA
        sta temp1         ;stores the contents of ACCA (the hexadecimal number 1
        ;(decimal 1)) in temp1 (the address $C0)
loop1:  lda temp1          ;loads binary 00000001 into ; the accumulator ACCA
        and porta         ;performs the logical AND between 00000001
        ;and the contents of port A and places the result in ACCA
        rora              ;rotate bit 0 into carry
        rora              ;rotate carry into bit 7
        sta porta         ;send bit7 to output port
        jmp loop1         ;jump to the label loop1 and repeat forever
        org $7FE         ;sets the address $07FE for the object code that follows
back1:  fdb start         ;forms double byte address constant start that defines
        ;where the program counter begins on reset

```

PRELAB:

Using Examples 1 and 2 as a guide, develop the following program: Use bit 7 of the port A as LED1 output and bit 0 of the port A as the switch S4 input. The program flashes the LED1 unless the S4 switch is pressed. When the S4 switch is pressed, the LED should be off. When it is pressed again, the LED1 should flash again, etc.

PROCEDURE:

A. Program from Example 1.

Enter and run the program from Example 1. See the result in PORTA, Variables Window (VARIABLES F8, Fig. 1) and on LED1, M68HC705JICS kit.

Note:after you press *F6* and see line

```
Debugger command:C:\ICS05J1A\FIRST1.S19
```

type in space and 2 so the line will be

```
Debugger command:C:\ICS05J1A\FIRST1.S19 2
```

and only after this press *Enter* (FIRST1 is an example of file's name).

B. Program from Example 2.

Enter and run the program from Example 2. See the result in PORTA, Variables Window (VARIABLES F8, Fig. 1) and on LED1, M68HC705JICS kit. Press the switch *S4*, M68HC705JICS kit and see the new result.

C. Program from prelab.

Enter and run the program from your prelab. See the result in PORTA, Variables Window (VARIABLES F8, Fig. 1) and on LED1, M68HC705JICS kit. Press the switch *S4*, M68HC705JICS kit and see the new result. Demonstrate the working program to the lab instructor and turn in the program listing.

Exercise

Write a program that flashes the LED whenever the button is pressed two times. When flashing, if the button is pressed once, the LED should stop flashing and remain unlit until another sequence of two button presses makes it flash again. This program is due at the beginning of next lab. Demonstrate the working program to your lab instructor and turn in the program listing. This is not a group assignment. All students should write their own programs and demonstrate them. The program should be clearly documented.