Experiment #1

ECE 367
Spring 2012

Shuaib Shameem
673551999

January 25th, 2012

TA: Yasaman Keshtkarjahromi

Schematic:

Attached to End of Report

Program Code:

```
; University of Illinois at Chicago, Dept. of Electrical and
Computer Engineering
; ECE 367 -Microprocessor-Based Design
; Three Bit Counter
; Version 2, Aug 27, 2011
; By Robert A. Becker

; PAY ATTENTION TO THE ALIGNMENT BELOW
; Labels start in the first column  (left most column = colunm 1)
; OP CODES are at column 9
; COMMENTS follow a ";" symbol
; Blank lines are allowed (Makes the code more readable)

; Define symbolic constants
PortT      EQU   $240       ;Define Register Locations
DDRT       EQU   $242
INITRG     EQU   $11
INITRM     EQU   $10
CLKSEL     EQU   $39
PLLCTL     EQU   $3A
CRGFLG     EQU   $37
SYNR       EQU   $34
REFDV      EQU   $35
COPCTL     EQU   $3C
TSCR1      EQU   $46
TSCR2      EQU   $4D
TIOS       EQU   $40
TCNT       EQU   $44
TC0        EQU   $50
TFLG1      EQU   $4E
;
; The ORG statment below would normally be followed by variable
definitions
; There are no variables needed for this project.
; THIS IS THE BEGINNING SETUP CODE
;
      ORG   $3800      ; Beginning of RAM for Variables
;
; The main code begins here. Note the START Label
;
      ORG   $4000      ; Beginning of Flash EEPROM
START LDS   #$3FCE           ; Top of the Stack
      SEI              ; Turn Off Interrupts
      MOVB  #$00, INITRG     ; I/O and Control Registers Start at
$0000
```

```
      MOVB  #$39, INITRM    ; RAM ends at $3FFF
;
; We Need To Set Up The PLL So that the E-Clock = 24MHz
;
      BCLR CLKSEL,$80 ; disengage PLL from system
      BSET PLLCTL,$40 ; turn on PLL
      MOVB #$2,SYNR   ; set PLL multiplier
      MOVB #$0,REFDV  ; set  PLL divider
      NOP        ; No OP
      NOP        ; NO OP
PLP   BRCLR CRGFLG,$08,PLP ; while (!(crg.crgflg.bit.lock==1))
      BSET CLKSEL,$80 ; engage PLL
;
;
;
      CLI        ; Turn ON Interrupts
;
; End of  setup code. You will always need the above setup code
for every experiment
;
      LDAA  #$FF  ; Make PortT Outbound
      STAA  DDRT
      LDAA  #$00  ; Start the count at 0
AG:   STAA  PortT ; Output the count to PortT
      JSR   DELAY ; Let's go wait one second
      INCA        ; Increment the count
      BNE   AG    ; Do again unless the count > 255
      STAA  PortT ; If we get here the output will go blank
      BRA   *     ; Stop the program (Branch here forever)
;
; We use some built-in timer functions to create an accurate
delay
;
DELAY PSHA             ; Save accumulator A on the stack
      LDY   #10        ; We will repeat this subroutine 10 times
      MOVB  #$90,TSCR1 ; enable TCNT & fast flags clear
      MOVB  #$06,TSCR2       ; configure prescale factor to 64
      MOVB  #$01,TIOS  ; enable OC0
      LDD   TCNT       ; Get current TCNT value
AGAIN ADDD  #37500           ; start an output compare operation
      STD   TC0        ; with 100 ms time delay
WAIT  BRCLR TFLG1,$01,WAIT  ; Wait for TCNT to catch up
      LDD   TC0        ; Get the value in TC0
      DBNE  Y,AGAIN          ; 10 X 100ms = 1 sec
      PULA             ; Pull A
      RTS
;
; End of counter code

; Define Power-On Reset Interrupt Vector

; AGAIN - OP CODES are at column 9
```

```
        ORG $FFFE ; $FFFE, $FFFF = Power-On Reset Int. Vector
Location
        FDB START ; Specify instruction to execute on power up

; End of Interrupt code

        END       ; (Optional) End of source code
```

Conclusion:

This project was extremely simple. The circuit design was complete within minutes of starting. The programming was supplied to us, so the code was completed even before we began. Pushing the code to the chip and getting the chip to run the code was just as simple as the rest of the lab was. In the end, the circuit and chip worked as expected. I feel extremely ready for the next and following experiments.