Khari Ollivierre
Lab 3
Section 1490

Pseudocode

Lab 3a

Define clock control/status and IOREG codes as constants

Define CLKEVOUT code for PORTC_6 output ("00000001")

BEGIN MAIN

    CALL CLKSEL

    SET PORTC_7 to output

    SET PORTCFG_CLKEVOUT to constant defined above

    WHILE true

    END WHILE

END MAIN

BEGIN CLKSEL

    SET CPU_CCP to IOREG (0xD8)

    WHILE OSC_STATUS is ~(status constant for desired clock)

    END WHILE

    SET OSC_CTRL to (control constant for desired clock)

    SET CLK_PSCTRL to (chosen prescalar settings)

    SET CLK_CTRL to (control constant for desired clock)

END CLKSEL

Lab 3b

Define clock control/status and IOREG codes as constants

DEFINE PERIOD_L as constant "0x00"

DEFINE PERIOD_H as constant "0xFF"

DEFINE C0_CTRL as constant "0x07"

BEGIN MAIN

    LOAD 4 registers used in CLKSEL with codes for 2 MHz clock

    CALL CLKSEL

    CALL CNTCFG_C0

    SET PORTC to output

    WHILE true

        OUTPUT TCCO_CNTL to PORTC

    END WHILE

END MAIN

BEGIN CLKSEL

    SET CPU_CCP to IOREG (0xD8)

    WHILE OSC_STATUS is ~(status constant for desired clock)

    END WHILE

    SET OSC_CTRL to (control constant for desired clock)

    SET CLK_PSCTRL to (chosen prescalar settings)

    SET CLK_CTRL to (control constant for desired clock)

END CLKSEL

BEGIN CNTCFG_C0

    LOAD PERIOD_L into TCC0_PERL

    LOAD PERIOD_H into TCC0_PERH

```
        LOAD C0_CTRL into TCC0_CTRLA
END CNTCFG_C0
DEFINE EBI_CTRL constant "0x41"
DEFINE EBI_CS0_CTRLA constant "0x42"
DEFINE EBI_CS0_CTRLB constant "0x06"
DEFINE EBI_CS0_BASEADDR constant "0x2000"
DEFINE TDATA constant "0xA5"
DEFINE TDATA_ADDR constant "0x04"

DEFINE PORTH pins as appropriate control signals

BEGIN MAIN
        CALL EBICONFIG
        SET CS0 to 0

        // Load value for writing
        LOAD TDATA into R18
        WHILE true
                LOAD TDATA_ADDR into Z  // Load high and low bytes individually
                CALL WRITE
                LOAD TDATA_ADDR into Z
                CALL READ           // Read operation results stored in R16
        END WHILE
END MAIN

BEGIN MEMADDRLD
        SET PORTK to ZL
        INC Z
END MEMADDRLD
```

```
BEGIN WRITE
        CALL MEMADDRLD
        SET PORTJ to R18    // Low byte to be written
        LSR R18                         // Mask out low byte
        CALL WRTGL
        CALL MEMADDRLD
        SET J[3-0] to R18      // High byte to be written
        CALL WRTGL
END WRITE

BEGIN WRTGL
        SET CS to 0
        SET ALE to 1
        SET WE to 0
        // Just in case
        NOP
        NOP
        NOP
        NOP
        NOP
        SET CS to 1
        SET ALE to 0
        SET WE to 0
END WRTGL

BEGIN READ
        CALL MEMADDRLD
        CALL RDTGL
```

```
    LOAD PORTJ into R16      // Low byte read
    CALL MEMADDRLD
    CALL RDTGL
    LOAD PORTJ into R17      // High byte read
    LSL R17                  // Shifted to its correct position
    ADD R16, R17             // Concatenate both bits
    CLR R17
END READ

BEGIN RDTGL
    SET CS to 0
    SET ALE to 0
    SET RE to 1
    NOP
    NOP
    NOP
    NOP
    SET CS to 1
    SET ALE to 0
    SET RE to 0
END RDTGL

BEGIN EBICONFIG
    SET EBI_CTRL to constant defined above
    SET EBI_CS0_CTRLA to constant defined above
    SET EBI_CS0_BASEADDR to constant defined above
    SET PORTH to output for control signals
    SET PORTK to output for address signal
```

```
        // Control signals predefined above
        SET CS to 1
        SET WE to 1
        SET RE to 1
        SET ALE to 0
        SET CS3 to 1
        SET CS2 to 1
        SET CS1 to 1
        SET CS0 to 1
END EBICONFIG
```

Lab 3d

// Reserved registers: R16 (Read register), R17 (temp register for reads), R18 (write register)

Define clock control/status and IOREG codes fo 32 MHz clock as constants

Define CLKEVOUT code for PORTC_6 output ("00000001")

DEFINE EBI_CTRL constant "0x41"

DEFINE EBI_CS0_CTRLA constant "0x42"

DEFINE EBI_CS0_CTRLB constant "0x06"

DEFINE EBI_CS0_BASEADDR constant "0x3000"

DEFINE PORTH pins as EBI control signals

DEFINE PERIOD_L as constant "0xA2"

DEFINE PERIOD_H as constant "0x07"

DEFINE C0_CTRL as constant "0x07"

DEFINE CCAFLAGS as "10000000"

BEGIN MAIN

    SET PORTA as input for switches

    SET PORTC as output for LEDs

    CLR Z   // For accessing external memory

    CALL CLKSEL

    CALL EBICONFIG

    CALL CNTCFG_C0

    CLR R19   // Register for counter flags

    WHILE true

        WHILE R19 != CCAFLAGS

```
                LOAD TCC0_INTFLAGS into R19
                R19 << 3                    // Mask out extra flag bits
            END WHILE
            LOAD PORTA into R18
            CALL WRITE                      // Write switch data to memory
            CLR Z                           // Reset Z to original memory location
            CALL READ                       // Read memory into R16
            STORE R18 on PORTC              // Output read data to LEDS
        END WHILE
END MAIN

BEGIN CLKSEL
        SET CPU_CCP to IOREG (0xD8)
        WHILE OSC_STATUS is ~(status constant for desired clock)
        END WHILE
        SET OSC_CTRL to (control constant for desired clock)
        SET CLK_PSCTRL to (chosen prescalar settings)
        SET CLK_CTRL to (control constant for desired clock)
END CLKSEL

BEGIN CNTCFG_C0
        LOAD PERIOD_L into TCC0_PERL
        LOAD PERIOD_L into TCC0_CCAL
        LOAD PERIOD_H into TCC0_PERH
        LOAD PERIOD_H into TCC0_CCAH
        LOAD C0_CTRL into TCC0_CTRLA
END CNTCFG_C0

BEGIN MEMADDRLD
```

```
        SET PORTK to ZL
        INC Z
END MEMADDRLD

BEGIN WRITE
        CALL MEMADDRLD
        SET PORTJ to R18    // Low byte to be written
        LSR R18                         // Mask out low byte
        CALL WRTGL
        CALL MEMADDRLD
        SET J[3-0] to R18      // High byte to be written
        CALL WRTGL
END WRITE

BEGIN WRTGL
        SET CS to 0
        SET ALE to 1
        SET WE to 0
        // Just in case
        NOP
        NOP
        NOP
        NOP
        NOP
        SET CS to 1
        SET ALE to 0
        SET WE to 0
END WRTGL
```

```
BEGIN READ
        CALL MEMADDRLD
        CALL RDTGL
        LOAD PORTJ into R16        // Low byte read
        CALL MEMADDRLD
        CALL RDTGL
        LOAD PORTJ into R17        // High byte read
        LSL R17                    // Shifted to its correct position
        ADD R16, R17               // Concatenate both bits
        CLR R17
END READ

BEGIN RDTGL
        SET CS to 0
        SET ALE to 0
        SET RE to 1
        NOP
        NOP
        NOP
        NOP
        SET CS to 1
        SET ALE to 0
        SET RE to 0
END RDTGL

BEGIN EBICONFIG
        SET EBI_CTRL to constant defined above
        SET EBI_CS0_CTRLA to constant defined above
        SET EBI_CS0_BASEADDR to constant defined above
```

SET PORTH to output for control signals

SET PORTK to output for address signal

// Control signals predefined above

SET CS to 1

SET WE to 1

SET RE to 1

SET ALE to 0

SET CS3 to 1

SET CS2 to 1

SET CS1 to 1

SET CS0 to 1

END EBICONFIG

Lab 3c