Martin Nguyen

ECE372

Dr. Hall

Project2\_Part 1

Code section:

**.text**

**.global** \_start

**.global** INT\_DIRECTOR

**\_start:**

@@@@@@@@SET UP THE STACK@@@@@@@@@@@@@@@@@@@@@@@@@@@

LDR R13,=STACK1 @Point to base of STACK for svc mod

**ADD** R13,R13,#0x1000 @Point to top of STACK

CPS #0x12 @Switch to IRQ mode

LDR R13,=STACK2 @Point to IRQ mode

**ADD** R13,R13,#0x1000 @Point to top of STACK

CPS #0x13 @Back to SVC mode

@==================================================

@Wake up I2C2 clock

LDR R1, =0x44E00044 @base address of CM\_PER\_I2C2\_CLKTRL

LDR R2, =0x02 @value to wake up the clock

STR R2,[R1] @store the value to the I2C2 clock

@==================================================

@Before enable the I2C2

@Setting up Prescale value for the clock to get 12Mhz

LDR R1, =0x4819C000 @base address of I2C2

**ADD** R2, R1, #0xB0 @offset of I2C2 CLock Prescaler Register

**MOV** R3, #0x3 @setting value to divide by 4

STR R3, [R2] @divide 48Mhz by 4 to obtain 12Mhz

@set up the Low time for I2C2\_SCLL

**ADD** R2, R1, #0xB4 @off set of I2C2\_SCLL register

**MOV** R3, #0x08 @ Value to set it to 400Kps. 8 in decimal and 0x08 in hex

STR R3, [R2] @store the value to I2C2\_SCLL

@set up the High time for I2C2\_SCHL

**ADD** R2, R1, #0xB8 @offset of I2C2\_SCHL register

**MOV** R3, #0x0A @value to set it to 400Kps. 10 in decimal and 0x0A in hex

STR R3, [R2] @store the value to I2C2\_SCHL

@Configure its own address

**ADD** R2, R1, #0xA8 @Address of I2C2\_OA

**MOV** R3, #0x00 @Value to reset it

STR R3, [R2] @store the value to I2C2\_OA

@Take the I2C module out of reset

**ADD** R2, R1, #0xA4 @address of I2C2\_CON

**MOV** R3, #0x8600 @enable the 15 bits to enable the module

STR R3, [R2]

@==================================================

@Initializing and setting up I2C2

@Switching Pin 19 and Pin 20 in BeagleBone to Mode 3 for I2C2\_SCL and I2C2\_SDA

LDR R1, =0x44E1097C @base address of UART\_RSTN

LDR R3, =0x33 @value to switch to mode 3, set the PullUp/PullDown enable and select PullUp, and Enable the receiver

STR R3, [R1] @switch to mode 3

LDR R1, =0x44E10978 @base address of UART\_CTSN

STR R3, [R1] @value to switch to mode 3, set the PullUp/PullDown enable and select PullUp, and Enable the receiver

@====================================================

@ Configure Slave Address and DATA counter register

@setting up DATA counter

LDR R1, =0x4819C000 @base address of I2C2

**ADD** R2, R1, #0x98 @offset of Data counter register

**MOV** R3, #0x02 @value to write 2 characters

STR R2, [R2] @store the value to count

@setting up the slave

**ADD** R2, R1, #0xAC @address of IC2C2 Slave register

LDR R4, [R2] @load the value to modify

**MOV** R5, #0x60 @value to have

STR R5, [R2] @store the value to the slave

@===================================================

@Transmission Procedure

LDR R1, =0x4819C000 @base address of I2C2

**ADD** R2, R1, #0x24 @address of I2C2\_IRQSTATUS\_RAW

LDR R3, =0x7FFF @set the XRDY to 0 to start transmitting data

STR R3, [R2] @store the value to the IRQSTATUS

@===================================================

BL POLL\_BB

BL START

@===================================================

@Sending out data

**SEND:**

BL DELAY

BL POLL\_XRDY

BL DELAY

@Send data to I2C2\_DATA

LDR R0,=0x4819C09C

**MOV** R2, #0x7D

STR R2, [R1]

@Clear the XRDY bit to send the next one

BL DELAY

BL DELAY

@===================================================

@Bus free for transmiting

**POLL\_BB:**

LDR R1, =0x4819C024 @base address of I2C2\_IRQSTATUS\_RAW

LDR R2, [R1] @load the value in

**CMP** R2, #0x1000 @check if the bit is clear or set

BEQ POLL\_BB @keep Polling if the Bus is busy

**MOV** PC, R14 @return

**POLL\_XRDY:**

LDR R0, =0x4819C024 @base address of I2C2\_IRQSTATUS\_RAW

LDR R2,[R0] @load in the address

TST R2, #0x10 @Check bit 4 for XRDY

BEQ POLL\_XRDY @if it not set then keep Polling

**MOV** PC, R14 @else return

**START:**

LDR R1, =0x4819C0A4 @base address of I2C2\_CON

LDR R2, =0x8603 @Turn on the STT and STP, STT = 1 and STP = 1, Start and STop

STR R2, [R1] @store the value to register

**MOV** PC, R14 @Return

**DELAY:**

LDR R1, =0x00040000 @delay for 0.2 seconds

**WAIT\_LOOP:**

SUBS R1,R1,#1

BNE WAIT\_LOOP

**MOV** PC, R14

**.data**

**.align** 2

**STACK1:** .rept 1024

**.word** 0x0000

.endr

**STACK2:** .rept 1024

**.word** 0x0000

.endr

.end

