@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

@ ECE 371 – PROGRAMMING PROJECT II

@----------------------------------------------------------------------------

@ USR LEDS AND GPIO

@ The program will turn the 4 BeagleBone Black USR LEDs on and off in a

@ rotating pattern like a theater marquee with delay loop timing.

@ The USR LEDS will rotate from USR LED 0 to USR LED 3 and go back to USR LED 0

@----------------------------------------------------------------------------

@ Hai Dang Hoang

@ Reference: Douglas V. Hall and Leela Yadlapalli

@ DECEMBER 18th, 2014

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

**.text**

**.global** \_start

**\_start:**

LDR R13, =STACK @ Create stack

ADDS R13,R13,#100 @ Point stack pointer at top of stack

@ Load value to turn off all three LEDs USR at bit 21,22,23 and 24 of GPIO1

LDR R0,=0x4804C000 @ Base address GPIO1 registers

LDR R7,=0x01E00000 @ GPIO1 21-24 off with GPIO1\_CLEARDATAOUT register

ADD R4,R0,#0x190 @ Make GPIO1\_CLEARDATAOUT register address

STR R7,[R4] @ Write to GPIO1\_CLEARDATAOUT register

@ Program GPIO1\_21-24 as outputs

ADD R5,R0,#0x0134 @ Make GPIO1\_OE register adress

LDR R6,[R5] @ Read current GPIO1\_OE Output Enable register

LDR R7,=0xFE1FFFFF @ Word to enable GPIO1\_21-24 as outputs

AND R6,R7,R6 @ Clear bits 21-24 ( Modify)

STR R6,[R5] @ Write to GPIO1 Output Enable register

@ Create a array storing status of LED\_Display

LDR R1,=LED\_Display @ Initialize pointer array for turning 4 USR LEDS on

MOV R5, #0x0 @ R5 Register(Index) control status of LED\_Display

LDR R7,=0x01E00000 @ Value for clear all 4 USR LEDS

@ Rotating 4 USR LEDS ( Loop forever )

**LOOP:**

BL LED\_on @ 4 USR LEDS run from right to left (GPIO1.21-> GPIO1.24)

BL Delay @ Call delay 2s Procedure

STR R7,[R4] @ Clear all 4 USR LEDS ( R4 current stores address of GPIO1\_CLEARDATAOUT)

CMP R5,#12 @ Check index, if index >12 -> index = 0

MOVEQ R5,#0x0 @ If equals -> Reset index = 0

ADDNE R5,R5,#4 @ If not, increment Pointer by 4

B LOOP @ Jump to LOOP

@ Procedure for rotating 4 USR LEDS

**LED\_on:**

STMFD R13!,{R2-R5,R14} @ Store uses registers on Stack

ADD R2,R0,#0x194 @ Load address of GPIO1\_SetDataOut

LDR R3,[R1,R5] @ Load value for turn in USR LED Depending on R5 (Index)

STR R3,[R2] @ Turn on led by storing the value into GPIO1\_SET\_DATA\_OUT

LDMFD R13!,{R2-R5,R14} @ Restore saved resigisters

MOV PC,R14 @ Return to mainline

@ Procedure for delaying 2s

**Delay:**

STMFD R13!,{R9,R14} @ Save uses registers

LDR R9,=0x00400000 @ Intialize delay loop counter

**NEXT:**

SUBS R9,R9,#0x1 @ Decrement loop counter

BNE NEXT @ Until loop counter equal 0

LDMFD R13!,{R9,R14} @ Restore values for saved registers

MOV PC, R14 @ Return to mainline

**.data**

**LED\_Display:** **.word** 0x00200000, 0x00400000, 0x00800000, 0x01000000 @ Values for turn in 4 USR LEDS

@ 0x00200000: USR1 GPIO1\_21 on

@ 0x00400000: USR2 GPIO1\_22 on

@ 0x00800000: USR3 GPIO1\_23 on

@ 0x01000000: USR4 GPIO1\_24 on

**STACK:** .rept 256

**.byte** 0x00

.endr

.end