Design Log

Phil Nevins
ECE 371 Microprocessor I
Design Project 2
11/19/2022

Part 1

Entry Date 11/19/2022 (Intro to project. Task list. Part 1 deliverables and goals)

Today, I will be reading through all of Part 1 for Design Project 2. Here is a summary of what is required within Part 1 of this design project, the deliverables, and tasks to be completed. Tasks include research / reading that must be done to understand how to design what is requested within the design specifications and what exactly our programs must achieve.

We are being tasked with determining which GPIO pins are connected to the 4 Usr LEDs and what logic level must be asserted to turn them on. This will be found in the BBB manual.

We are also tasked with working through Hall Ch4, specifically the parts that describe the AM3358 GPIO pins and the memory mapped registers that control them.
We need to determine which registers control the GPIO pins connected to these LEDs.

Once we fully understand which GPIO pins and logic levels turn on the 4 USR LEDs, which memory mapped registers control them, our first task is designing a program that will turn on Usr LED 3 for one second and turn it off for one second and repeat these steps. (From the design specifications: Note that to single step through the program easily you can initially use a very small delay constant, so you don't have to single step forever to get from LED on to LED off.)

When this program works, we will make a new copy of the program to continue onto the next task, saving the previous program that turns on Usr LED 3.

The second task is to design a program that will light LED3 for one second, light LED2 for one second, light LED1 for one second, light LED 0 for one second, then cycle back to LED1, then LED2, then LED3. This is known as a "cyclone eye". This is something common in any scrolling display, like the marquee in a theater or the scrolling LED text walls in sports arenas.

Once the second task is completed, we need to make a new copy of the program to continue onto Part 2, saving the previous program that operates a cyclone eye across the Usr LEDs 1 – 4.

The deliverables for Part 1 of Design Project 2 include a PDF file that is comprised of the design log, pseudocode for the low and high level algorithms, and a copy of the .s files converted to PDF

The gap in the design log is due to an illness that had me out of commission from 11/20 – 11/23

Entry 11/24/2022 (Research summary, general idea / algorithm)

I have read through all the required material and studied it thoroughly. In this design log entry, I will be summarizing what I learned and what is required to meet the design specifications for our programs in Part 1. I will also outline the high-level algorithm for achieving the design specifications goals, then write the low-level algorithm that will be translated into assembly language.

Table 8. User LED Control Signals/Pins

LED	GPIO SIGNAL	PROC PIN
USR0	GPIO1_21	V15
USR1	GPIO1_22	U15
USR2	GPIO1_23	T15
USR3	GPIO1_24	V16

A logic level of "1" will cause the LEDs to turn on.

Source: Github Link for BBB Reference Manual

The method of turning on the LEDs, we need to do a couple of things first.
1. Enable the CLKs for GPIO1 (CM_PER_GPIO1_CLKCTRL Register) must be sent 0x02. We were shown examples of this in the textbook.

2. Program GPIO1 21-24 (or 24 for just LED3) as outputs. We do this by sending all HIGHs except for the specific locations we want to turn enable as outputs, which is sent LOWs. That is where we got the Enable Output words from. Each of those values is what you would send to the GPIO1_OE Register, to enable each GPIO1_21-24 as outputs. If you do not do this, the program will not operate properly.

3. Now we can write a HIGH to the SETDATAOUT location to turn ON each LED. You need to send all LOWs except for the 21st – 24th bit, which needs to be a HIGH. Whichever bit you send a high to at the SETDATAOUT location, will turn on that LED. That is where we got the Write High words from.

4. You can do the exact same thing to turn the LEDs OFF, but instead of the SETDATAOUT location, you send the same Write High word to the CLEARDATAOUT location. These locations, when sent a 1, triggers it to do the action. So, sending a 1 in any bit position to the CLEARDATAOUT, you will clear the data in that bit position. So, if you send 0x00000001 to

SETDATAOUT, you will set the GPIO -1 to HIGH. If you send 0x00000004, you will set GPIO -3 HIGH, and so on.

GPIO1_21 (LED0)
Enable Output – FFDFFFFF
Write High – 0x00200000

GPIO1_22 (LED1)
Enable Output - FFBFFFFF
Write High – 0x00400000

GPIO1_23 (LED2)
Enable Output - FF7FFFFF
Write High – 0x00800000

GPIO1_24 (LED3)
Enable Output – FEFFFFFF
Write High – 0x01000000

Delay Loop 1 Second
0x00333333
(This was derived from the example in the book, where the delay is 5 seconds. So, we take 0x00FFFFF and divide it by 5, to get 0x00333333)

GPIO1 Clock Addr - 0x44E000AC
GPIO1_SETDATAOUT Addr - 0x4804C194
GPIO1_CLEARDATAOUT Addr - 0x4804C190

GPIO1 21-24
Enable Output – 0xFE1FFFFF
This enables all 4 LEDs, which is what we will use for the cyclone eye.

General Algorithm Cyclone Eye

We will achieve this by having two loops, Loop0123 and Loop3210. Loop0123 will end and go straight into Loop3210. At the end of Loop3210, it will branch back to Loop0123. This will enable us to utilize these loops in later programs if we need to turn the LEDs on in either pattern.

Now that we know all these specific details, we can build our pseudocode for what we are asked to do in the design specifications. Which is turning LED3 on for one second, off for one second, over and over. The second is “cyclone eye”, which means LED 0 turns on, then LED1, then LED2, then LED3, then LED2, then LED1, then it repeats. Each LED stays on for 1 second using a delay timing loop. So, we will visually see the LEDs go in this pattern, 012321012321 and repeat this ‘forever’.

Entry 11/25/2022 (Writing the code, debugging and final notes)

Today, I am writing the code for part 1 of Design Project 2. In the initial steps of just getting LED3 to turn on and off, I was having issues getting the LED to turn off and then I realized I was writing to the SETDATAOUT instead of the CLEARDATAOUT. I put in the wrong address for the CLEARDATAOUT. Fixing this made the program execute correctly.

The cyclone eye went smoothly.

It seemed inefficient to branch to the next function when it is right after the last inline.

Example: LED3ON, at the end, branches to LED3OFF. Without this branch command, the program will do it automatically since they are inline.

When I attempted to remove this during optimization, it caused the LED to stay on for ~1.3 seconds. So, it is required to make the timing of 1 second on, 1 second off.

Pseudocode for High Level

Single LED

Activate LED

Repeat
    Turn LED3 on
        Repeat
        Until 1 second passed
    Turn LED3 off
Forever

Cyclone Eye

Activate LEDs

Repeat
    Turn LED3 on
        Repeat
        Until 1 second passed
    Turn LED3 off
    Turn LED2 on
        Repeat
        Until 1 second passed
    Turn LED2 off

Turn LED1 on
Repeat
Until 1 second passed
Turn LED1 off
Turn LED0 on
Repeat
Until 1 second passed

Turn LED0 off
Turn LED1 on
Repeat
Until 1 second passed
Turn LED1 off
Turn LED2 on
Repeat
Until 1 second passed
Turn LED2 off
Forever

Pseudocode for Low Level

Single LED

Enable GPIO1 clocks using value 0x02, at the address 0x44E000AC

Set GPIO1_24 to output using 0xFEFFFFFF via GPIO1_OE Register
At address 4804C134 using the 3-step method

LED ON FUNCTION:
Load word to target GPIO1_24 (LED3), 0x01000000
Load address of GPIO1_SETDATAOUT, 0x4804C194
Write word to SETDATAOUT (This turns ON LED3)
Load delay timer value 0x00333333
BL to DELAY FUNCTION
B to LED OFF FUNCTION

LED OFF FUNCTION:
Load word to target GPIO1_24 (LED3), 0x01000000
Load address of GPIO1_CLEARDATAOUT, 0x4804C190
Write word to CLEARDATAOUT (This turns OFF LED3)
Load delay timer value 0x00333333
BL to DELAY FUNCTION
B to LED ON FUNCTION

DELAY FUNCTION:
Subtract 1 from delay timer value
Repeat until 0
Move LR to program counter to get back to LED function

Cyclone Eye

Enable GPIO1 clocks using value 0x02, at the address 0x44E000AC

Set GPIO1_21-24 to output using 0xFE1FFFFF via GPIO1_OE Register
At address 4804C134 using the 3-step method

LED3ON FUNCTION:
Load word to target GPIO1_24 (LED3), 0x01000000
Load address of GPIO1_SETDATAOUT, 0x4804C194
Write word to SETDATAOUT (This turns ON LED3)
Load delay timer value 0x00333333
BL to DELAY FUNCTION
B to LED3OFF FUNCTION

LED3OFF FUNCTION:
Load word to target GPIO1_24 (LED3), 0x01000000
Load address of GPIO1_CLEARDATAOUT, 0x4804C190
Write word to CLEARDATAOUT (This turns OFF LED3)
Load delay timer value 0x00333333
BL to DELAY FUNCTION
B to L2R-LED2ON FUNCTION

L2R-LED2ON FUNCTION:
Load word to target GPIO1_23 (LED2), 0x00800000
Load address of GPIO1_SETDATAOUT, 0x4804C194
Write word to SETDATAOUT (This turns ON LED2)
Load delay timer value 0x00333333
BL to DELAY FUNCTION
B to L2R-LED2OFF FUNCTION

L2R-LED1OFF FUNCTION:
Load word to target GPIO1_23 (LED2), 0x00800000
Load address of GPIO1_CLEARDATAOUT, 0x4804C190
Write word to CLEARDATAOUT (This turns OFF LED2)
Load delay timer value 0x00333333
BL to DELAY FUNCTION
B to L2R-LED1ON FUNCTION

L2R-LED1ON FUNCTION:
Load word to target GPIO1_22 (LED1), 0x00400000
Load address of GPIO1_SETDATAOUT, 0x4804C194
Write word to SETDATAOUT (This turns ON LED3)
Load delay timer value 0x00333333
BL to DELAY FUNCTION
B to L2R-LED1OFF FUNCTION

L2R-LED1OFF FUNCTION:
Load word to target GPIO1_22 (LED1), 0x00400000
Load address of GPIO1_CLEARDATAOUT, 0x4804C190
Write word to CLEARDATAOUT (This turns OFF LED3)
Load delay timer value 0x00333333
BL to DELAY FUNCTION
B to LED0ON FUNCTION

LED0ON FUNCTION:
Load word to target GPIO1_21 (LED0), 0x00200000
Load address of GPIO1_SETDATAOUT, 0x4804C194
Write word to SETDATAOUT (This turns ON LED3)
Load delay timer value 0x00333333
BL to DELAY FUNCTION
B to LED0OFF FUNCTION

LED0OFF FUNCTION:
Load word to target GPIO1_21 (LED0), 0x00200000
Load address of GPIO1_CLEARDATAOUT, 0x4804C190
Write word to CLEARDATAOUT (This turns OFF LED3)
Load delay timer value 0x00333333
BL to DELAY FUNCTION
B to R2L_LED1ON FUNCTION

**This is where it reverses direction

R2L-LED1ON FUNCTION:
Load word to target GPIO1_22 (LED1), 0x00400000
Load address of GPIO1_SETDATAOUT, 0x4804C194
Write word to SETDATAOUT (This turns ON LED3)
Load delay timer value 0x00333333
BL to DELAY FUNCTION
B to R2L-LED1OFF FUNCTION

R2L-LED1OFF FUNCTION:
Load word to target GPIO1_22 (LED1), 0x00400000
Load address of GPIO1_CLEARDATAOUT, 0x4804C190

Write word to CLEARDATAOUT (This turns OFF LED3)
Load delay timer value 0x00333333
BL to DELAY FUNCTION
B to R2L-LED2ON FUNCTION

R2L-LED2ON FUNCTION:
Load word to target GPIO1_23 (LED2), 0x00800000
Load address of GPIO1_SETDATAOUT, 0x4804C194
Write word to SETDATAOUT (This turns ON LED3)
Load delay timer value 0x00333333
BL to DELAY FUNCTION
B to R2L-LED1OFF FUNCTION

R2L-LED2OFF FUNCTION:
Load word to target GPIO1_23 (LED2), 0x00800000
Load address of GPIO1_CLEARDATAOUT, 0x4804C190
Write word to CLEARDATAOUT (This turns OFF LED3)
Load delay timer value 0x00333333
BL to DELAY FUNCTION
B to LED3ON FUNCTION

DELAY FUNCTION:
Subtract 1 from delay timer value
Repeat until 0
Move LR to program counter to get back to LED function

```
@Phil Nevins
@ECE 371 Microprocessor I
@Design Project 2, Part 1, Single LED
@This program will turn USR3 LED (GPIO1_24) on for one second and off for one second
@This pattern will be repeated forever
@Program uses R0-2, R5-7 and R9-10

.text
.global _start
_start:

@Initiliaze CLK
MOV R9, #0x02                   @Value to enable CLKs for GPIO modules
LDR R10, =0x44E000AC            @Addr of CM_PER_GPIO1_CLKCTRL Register
STR R9, [R10]                   @Write 0x02 to CLKCTRL Register

@Program GPIO1_24 as output
LDR R0, =0xFEFFFFFF             @Load word to program GPIO1_24 to output
LDR R1, =0x4804C134             @Addr of GPIO1_OE Register
LDR R2, [R1]                @Read GPIO1_OE Register
AND R2, R2, R0                  @Modify word read in with R0
STR R2, [R1]                @Write back to GPIO1_OE Register

LED3ON:                             @Turn on LED3 Function
    MOV R5, #0x01000000         @Load word to target GPIO1_24
    LDR R6, =0x4804C194         @Load addr of GPIO1_SETDATAOUT
    STR R5, [R6]            @Write to GPIO1_SETDATAOUT (This turns LED ON)
    LDR R7, =0x00333333         @Load Delay Timer value (one second)
    BL Delay1Sec            @Branch to delay timer function
    B LED3OFF                   @Branch to LED3OFF

LED3OFF:                            @Turn off LED3 Function
    MOV R5, #0x01000000         @Load word to target GPIO1_24
    LDR R6, =0x4804C190         @Load addr of GPIO1_CLEARDATAOUT
    STR R5, [R6]            @Write to GPIO1_CLEARDATAOUT (This turns LED OFF)
    LDR R7, =0x00333333         @Load Delay Timer value (one second)
    BL Delay1Sec            @Branch to delay timer function
    B LED3ON                    @Branch to LED3ON

Delay1Sec:                          @One Second Delay Loop
    SUBS R7, R7, #1             @Subtract 1 from delay timer value
    BNE Delay1Sec               @Loop until delay timer value is 0
    MOV PC, LR                  @Branch back to LEDON/OFF

.end
```

```
@Phil Nevins
@ECE 371 Microprocessor I
@Design Project 2, Part 1, Cyclone Eye
@LED3 turns on, then LED2, then LED1, then LED0, then LED1, then LED2,
@then it repeats. Each LED stays on for 1 second using a delay timing loop
@Program uses R0-2, R5-7 and R9-10

.text
.global _start
_start:

@Initiliaze CLK
MOV R9, #0x02                   @Value to enable CLKs for GPIO modules
LDR R10, =0x44E000AC            @Addr of CM_PER_GPIO1_CLKCTRL Register
STR R9, [R10]                   @Write 0x02 to CLKCTRL Register

@Program GPIO1_21-24 as output
LDR R0, =0xFE1FFFFF             @Load word to program GPIO1_21-24 to output
LDR R1, =0x4804C134             @Addr of GPIO1_OE Register
LDR R2, [R1]                @Read GPIO1_OE Register
AND R2, R2, R0                  @Modify word read in with R0
STR R2, [R1]                @Write back to GPIO1_OE Register

@LED3
LED3ON:                             @Turn on LED3 Function
    MOV R5, #0x01000000         @Load word to target GPIO1_24
    LDR R6, =0x4804C194         @Load addr of GPIO1_SETDATAOUT
    STR R5, [R6]            @Write to GPIO1_SETDATAOUT (This turns LED ON)
    LDR R7, =0x00333333         @Load Delay Timer value (one second)
    BL Delay1Sec            @Branch to delay timer function
    B LED3OFF                   @Branch to L2R_LED3OFF

LED3OFF:                            @Turn off LED3 Function
    MOV R5, #0x01000000         @Load word to target GPIO1_24
    LDR R6, =0x4804C190         @Load addr of GPIO1_CLEARDATAOUT
    STR R5, [R6]            @Write to GPIO1_CLEARDATAOUT (This turns LED OFF)
    LDR R7, =0x00333333         @Load Delay Timer value (one second)
    BL Delay1Sec            @Branch to delay timer function
    B L2R_LED2ON                @Branch to L2R_LED2ON

@LED2
L2R_LED2ON:                         @Turn on LED2 Function
    MOV R5, #0x00800000         @Load word to target GPIO1_23
    LDR R6, =0x4804C194         @Load addr of GPIO1_SETDATAOUT
    STR R5, [R6]            @Write to GPIO1_SETDATAOUT (This turns LED ON)
    LDR R7, =0x00333333         @Load Delay Timer value (one second)
    BL Delay1Sec            @Branch to delay timer function
    B L2R_LED2OFF               @Branch to L2R_LED2OFF

L2R_LED2OFF:                        @Turn off LED2 Function
    MOV R5, #0x00800000         @Load word to target GPIO1_23
    LDR R6, =0x4804C190         @Load addr of GPIO1_CLEARDATAOUT
    STR R5, [R6]            @Write to GPIO1_CLEARDATAOUT (This turns LED OFF)
```

```
	LDR R7, =0x00333333		@Load Delay Timer value (one second)
	BL Delay1Sec		@Branch to delay timer function
	B L2R_LED1ON			@Branch to L2R_LED1ON

@LED1
L2R_LED1ON:				@Turn on LED1 Function
	MOV R5, #0x00400000		@Load word to target GPIO1_22
	LDR R6, =0x4804C194		@Load addr of GPIO1_SETDATAOUT
	STR R5, [R6]		@Write to GPIO1_SETDATAOUT (This turns LED ON)
	LDR R7, =0x00333333		@Load Delay Timer value (one second)
	BL Delay1Sec		@Branch to delay timer function
	B L2R_LED1OFF			@Branch to L2R_LED1OFF

L2R_LED1OFF:				@Turn off LED1 Function
	MOV R5, #0x00400000		@Load word to target GPIO1_22
	LDR R6, =0x4804C190		@Load addr of GPIO1_CLEARDATAOUT
	STR R5, [R6]		@Write to GPIO1_CLEARDATAOUT (This turns LED OFF)
	LDR R7, =0x00333333		@Load Delay Timer value (one second)
	BL Delay1Sec		@Branch to delay timer function
	B LED0ON			@Branch to L2R_LED0ON

@LED0
LED0ON:				@Turn on LED0 Function
	MOV R5, #0x00200000		@Load word to target GPIO1_21
	LDR R6, =0x4804C194		@Load addr of GPIO1_SETDATAOUT
	STR R5, [R6]		@Write to GPIO1_SETDATAOUT (This turns LED ON)
	LDR R7, =0x00333333		@Load Delay Timer value (one second)
	BL Delay1Sec		@Branch to delay timer function
	B LED0OFF			@Branch to L2R_LED0OFF

LED0OFF:				@Turn off LED0 Function
	MOV R5, #0x00200000		@Load word to target GPIO1_21
	LDR R6, =0x4804C190		@Load addr of GPIO1_CLEARDATAOUT
	STR R5, [R6]		@Write to GPIO1_CLEARDATAOUT (This turns LED OFF)
	LDR R7, =0x00333333		@Load Delay Timer value (one second)
	BL Delay1Sec		@Branch to delay timer function
	B R2L_LED1ON			@Branch to R2L_LED1ON

@This is where it reverses direction
@LED1
R2L_LED1ON:				@Turn on LED1 Function
	MOV R5, #0x00400000		@Load word to target GPIO1_22
	LDR R6, =0x4804C194		@Load addr of GPIO1_SETDATAOUT
	STR R5, [R6]		@Write to GPIO1_SETDATAOUT (This turns LED ON)
	LDR R7, =0x00333333		@Load Delay Timer value (one second)
	BL Delay1Sec		@Branch to delay timer function
	B R2L_LED1OFF			@Branch to R2L_LED1OFF

R2L_LED1OFF:				@Turn off LED1 Function
	MOV R5, #0x00400000		@Load word to target GPIO1_22
	LDR R6, =0x4804C190		@Load addr of GPIO1_CLEARDATAOUT
	STR R5, [R6]		@Write to GPIO1_CLEARDATAOUT (This turns LED OFF)
	LDR R7, =0x00333333		@Load Delay Timer value (one second)
	BL Delay1Sec		@Branch to delay timer function
```

```
    B R2L_LED2ON                @Branch to R2L_LED2ON

@LED2
R2L_LED2ON:                         @Turn on LED2 Function
    MOV R5, #0x00800000     @Load word to target GPIO1_23
    LDR R6, =0x4804C194     @Load addr of GPIO1_SETDATAOUT
    STR R5, [R6]            @Write to GPIO1_SETDATAOUT (This turns LED ON)
    LDR R7, =0x00333333     @Load Delay Timer value (one second)
    BL Delay1Sec            @Branch to delay timer function
    B R2L_LED2OFF               @Branch to R2L_LED2OFF

R2L_LED2OFF:                        @Turn off LED2 Function
    MOV R5, #0x00800000     @Load word to target GPIO1_23
    LDR R6, =0x4804C190     @Load addr of GPIO1_CLEARDATAOUT
    STR R5, [R6]            @Write to GPIO1_CLEARDATAOUT (This turns LED OFF)
    LDR R7, =0x00333333     @Load Delay Timer value (one second)
    BL Delay1Sec            @Branch to delay timer function
    B LED3ON                    @Branch to R2L_LED3ON


Delay1Sec:                      @One Second Delay Loop
    SUBS R7, R7, #1             @Subtract 1 from delay timer value
    BNE Delay1Sec               @Loop until delay timer value is 0
    MOV PC, LR                  @Branch back to LEDON/OFF

.end
```

By signing this statement, I affirm that I did not give any help to any other person, did not receive any help from any other person, except TA and Instructor and did not obtain any information from the Internet or other sources.

Signature:_____Philip A Nevins_____ 

Date:_____11/26/2022_____