EGR 322 Microcontroller System Design

Lab 7 – Reference Multiplexer and Programmable Gain Amplifier

Prepared for: Dr. Foist

Christopher Parisi

College of Engineering
California Baptist University

04/03/12

Summary

This lab teaches students about the analog user modules. Students will study ground and reference signals then use the programmable gain amplifier. This lab will instruct students on how to implement this amplifier in the code as well as with the PSoC designer program.

Design

Exercise 1:
I first began by implementing the RefMux User Module. I followed the steps in the lab handout and created a new project that used the (Vdd/2+/-(Vdd/2) RefMux. I placed this module in ACB00 and set it to Reference Select and AGND. Once this was finished, I opened the main.asm and added

```
mov A, RefMux_HIGHPOWER
call RefMux_Start
```

to the program. I generated, built. And downloaded the code to my PSoC Evall board. I then used a multimeter to measure the voltage of the reference signal and corresponding values. After measuring the voltage, I changed the Ref Selection parameter to (2BandGap)+/-BandGap and reinstalled my code. The data I collected is shown below.

Ref Selection	**AGND (V)**	**RefHi (V)**	**RefLo (V)**
(Vdd/2)+/-(Vd/2)	2.5V	4.7V	0V
(2BandGap)+/-BandGap	2.6V	3.9V	1.3V

Exercise 2:
I then copied my lab7a project and renamed it to lab7b. I also changed Ref Mux to (2BandGap)+/-BandGap. I then added a PGA User Module and placed it in ACB01 and changed its input to AnalogColumn_InputSelect_1, its gain to 2.000, its reference to VSS and the AnlogBus to AnlogOutBus_1. I then added

```
mov A, PGA_HIGHPOWER
call PGA_Start
```

to the main.asm file and installed the program onto my Evall Board. A DC power supply was used to give the input and I then measured the input voltage and output voltage using a multimeter. The data I collected is shown below.

Desired Input Voltage (V)	**Measured Input Voltage (V)**	**Measured Output Voltage (V)**
0.5	0.49	0.98
1	1.00	1.99

Exercise 3:
For Exercise 3, I created a project that uses both a PGA and a sleep timer interrupt. I programmed the gain to increment every one second with the given gain values in the lab manual. For my project, I set the following paramters:

Ref Mux:	(2BandGap)+/-BandGap
Sleep_Timer:	1_Hz
PGA Input:	REFLO
Reference:	VSS
AnalogBus:	AnalogOutBus_0

I connected a wire from output pin P0[3] to LED1. The code I wrote for this program is shown below. After downloading the program, I powered the PSoC Evall Board and observed as the LED became brighter every one second. This corresponds to the increased gain caused by the sleep timer interrupt.

```
;-------------------------------------------------------------------------
; Assembly main line
;-------------------------------------------------------------------------

include "m8c.inc"        ; part specific constants and macros
include "memory.inc"     ; Constants & macros for SMM/LMM and Compiler
include "PSoCAPI.inc"    ; PSoC API definitions for all User Modules

export _main
export bShadow

area bss(RAM)
      bShadow: BLK 1

area text(ROM,REL)

_main:

      mov [bShadow], 0

      M8C_EnableGInt
      or reg[INT_MSK0], 0x40

    mov A, PGA_G0_75
      call PGA_SetGain
      mov A, PGA_HIGHPOWER
      call PGA_Start

loop:
     jmp loop




;-------------------------------
; Code added to boot.asm and boot.tpl
;-------------------------------

org   64h                       ;Sleep Timer Interrupt Vector
           ljmp SleepTimerISR
```

```
;-------------------------------
; SleepTimer ISR
;-------------------------------
include "m8c.inc"
include "memory.inc"
include "PSoCAPI.inc"

export SleepTimerISR

SleepTimerISR:

      push A
      inc [bShadow]
      and [bShadow], %00000111
      mov A, PGA_OFF
      call PGA_SetPower

      Gain_0_75:
      cmp [bShadow], %00000000
      jnz Gain_1_00
      mov A, PGA_G0_75
      jmp Return

      Gain_1_00:
      cmp [bShadow], %00000001
      jnz Gain_1_23
      mov A, PGA_G1_00
      jmp Return

      Gain_1_23:
      cmp [bShadow], %00000010
      jnz Gain_1_46
      mov A, PGA_G1_23
      jmp Return

      Gain_1_46:
      cmp [bShadow], %00000011
      jnz Gain_1_78
      mov A, PGA_G1_46
      jmp Return

      Gain_1_78:
      cmp [bShadow], %00000100
      jnz Gain_2_27
      mov A, PGA_G1_78
      jmp Return

      Gain_2_27:
      cmp [bShadow], %00000101
      jnz Gain_2_67
      mov A, PGA_G2_27
      jmp Return

      Gain_2_67:
      cmp [bShadow], %00000110
      jnz Gain_3_20
      mov A, PGA_G2_67
```

```
    jmp Return

    Gain_3_20:
    cmp [bShadow], %00000111
    jnz Gain_0_75
    mov A, PGA_G3_20

    Return:
    call PGA_SetGain
    mov A, PGA_HIGHPOWER
    call PGA_SetPower
    pop A
  reti
```

Discussion

One problem I encountered during this lab was having the code compile for exercise 3 but the LED not lighting up. I compared my code with the professor's code to confirm it was correct. I restarted and created a new project placing the PGA User Module in the correct place and copying my code to this new project. I also noticed the was a reti command after the ljmp in the boot.asm file. Once I got rid of reti and redownloaded everything, the LED lit up like it was supposed to.

Conclusion

This lab has taught me how to use the PSoC designer's User Modules. I have learned how to implement them into my project and how to use their different reference signals. I have also learned about the programmable gain amplifier (PGA). I know how to give the PGA different gains and also how to change these gains within a program. This lab has given me the knowledge necessary for understanding the PSoC in more detail.