SC2107 Lab1 Assignment Sheet (to be submitted to NTULearn before next lab)

Name: Chua Qin Di Lab Group: SCEB Date: 29/8/2024

1. Section 7.4. Write one C statement to set bit 7 and 5 of P1SEL0 register, keeping the rest of the bits in the register unchanged.

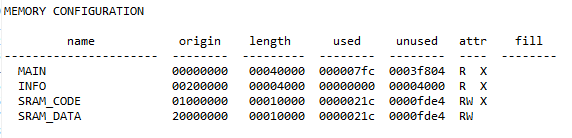
* Answer:   
  P1SEL0 |= (1 << 7) | (1 << 5);

1. Section 7.4. Write C statement(s) to extract bit 6 and 5 of variable ‘x’ and right align these two bits. Masked off all other bits in variable ‘x’. e.g. if ‘x’ has a value 1101 0111b initially, it should have a value of 0000 0010b after executing the C statement.
   * Answer:  
     x = (x & 0x60) >> 5;
2. Section 7.4. Why do we need to declare the P1IN register, the register that contain the status of the processor Port1 GPIO input pin logic with a ‘volatile’ keyword qualifier?
   * Answer:   
     The “volatile” keyword informs the compiler and optimizer that variables involving the data in the P1IN register may change outside the context of the program i.e. hardware changes.  
     This will prevent codes and functions involving these data from being removed/optimized.
3. Section 7.5. Why do we use SDIV instead of UDIV when calculating the Distance D? Or does it really matter whether SDIV or UDIV is used for this case?
   * Answer:   
     SDIV does signed division while UDIV does unsigned division. In the event of negative values, UDIV would give rise to unexpected results as it assumes that the values are unsigned. In this program, it would not matter as the convert sub-routine will return distance 800mm for values of n less than 2552 using CMP and BLT instructions before SDIV.
4. Section 7.5. What is saved into the LR register when the calling routine calls “BL Convert”? What command is used to return from the sub-routine to the calling routine?
   * Answer:

The address of the next instruction after the convert subroutine is saved into the LR register.

“BX LR” is used to return

1. Section 7.5. If a function has 4 input parameters, which registers does the calling routine used to pass these parameters to the function according to AAPCS?
   * Answer:  
     R0, R1, R2, R3
2. Section 7.6. What data content is loaded into R1 by the instruction “ldr r1, [pc, #0x2e4]”? Just the expression will do, need not give the exact value since the offset in your code may be different.
   * Answer:  
     Address of P1SEL0 is loaded in R1
3. Section 7.7. The Memory Section “MAIN” correspond to the On-Chip Flash Memory in MSP432. How much on-chip flash memory is available for future code development? Cut and paste the screen shot of the relevant content in the map file and highlight where you extract your answer from. Hint: Check the map file.
   * Answer:   
     0xfde4 bytes is left for further code development.



1. Section 7.7. Which software section are code allocated to by default? Which file consumes the largest code size in this project? Hint: check the map file.
   * Answer:   
     Code is allocated to .text section by default.  
     “system\_msp432p401r.obj” consumes the largest code size at 0x32c bytes.
2. Section 7.7. From the map file, what is the starting address of Port2\_Init()? Compare with the address you see in the Disassembly Window, are they the same? If not, why?
   * Answer:   
     Port2\_Init starting address is at 0x43b.  
     The starting address in the disassembly window is differs by 1 bit.   
     For ARM processors, the LSB of the target address indicates the instruction state when a branch instruction is executed.  
     If the LSB is 1, the processor switches to Thumb state and the actual starting address is PC-1