|  |  |  |
| --- | --- | --- |
| BLG351E  Experiment 1 “Basic Assembly Coding”  REPORT | CRN | 12633 |
| Group | G8 |
| Name #1 | Furkan Salık |
| Name #2 | Eyüpcan Çakır |
| Name #3 | Ahmet Fatih Akcan |
| Name #4 | Muhammed Yavuz Köseoğlu |
| Q1) (40 pts.) How to use PORT1 and PORT2 for I/O operations? Summarize your findings from the experiment. | | |
| First, directions must be set, P1DIR and P2DIR specify whether each 8 pins in each port is input or output. 0 is for input, and 1 is for output. Then, P1IN, P1OUT, P2IN, P2OUT is set by taking the directions into account. These registers and direction registers are 8 bit and *nth* bit specifies the state of the *nth* pin (on or off).  For example, if we want to use the pins 1-4 as inputs, and 5-8 as outputs in PORT1, we would set the direction register as following:  P1DIR: 11110000 (Bits 1-4 are set to 0, which is input, and 5-8 are set to 1, which is output)  If we want to set the outputs as high, we would set P1OUT register as following:  P1OUT: 11110000 (Pins 5-8 of the PORT1 are set as high)  All these port registers are set using absolute addressing. For example, **‘mov #11110000b, &P1OUT’.** | | |
| Q2) (30 pts.) Assume that you used the following code to solve Part 2. How the bits of Status Register changes after each operation? | | |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | V | N | Z | C | | SetupP1 mov.b #11111111b, &P1DIR | - | - | - | - | | mov.b #11111111b, &P2DIR | - | - | - | - | | start mov.b #00000001b, &P1OUT | - | - | - | - | | mov.b #00000001b, &P2OUT | - | - | - | - | | mov.b #0d, R6 | - | - | - | - | | MainLoop mov.w #050000, R15 | - | - | - | - | | L1 dec.w R15 | 0(1 when R15 equal 0x8000h) | 1(0 after R15 reach 0x7FFFh) | 0 (1 after the last execution in loop) | 0 | | jnz L1 | - | - | - | - | | rla.b &P1OUT | 0 | 0 | 0 | 0 | | rla.b &P2OUT | 0 | 0 | 0 | 0 | | inc.b R6 | 0 | 0 | 0 | 0 | | cmp #04d, R6 | 0 | 1 (0 after the last execution in loop) | 0 (1 after the last execution in loop) | 0 | | jeq start | - | - | - | - | | jmp MainLoop | - | - | - | - | | | |
| Q3) (30 pts.) What is the difference between **mov.b #00001111b, &P1OUT**, **bis.b #00001111b, &P1OUT** and **bic.b #00001111b, &P1OUT** and **bit.b #00001111b, &P1OUT**? Let the initial value of P1OUT be 10101010. What will be the new value after each operation? | | |
| **mov** sets the destination to given value (src -> dst), **bis** is basically an OR operation (src OR dst -> dst), **bic** clears the bits in destination whose value in source is 1 ( AND dst -> dst), **bit** is an AND operation and is used to test the bits set in source value using zero flag (src AND dst).  So first three of the instructions perform assignment operations and do not affect status register, however bit test instruction does not perform any assignment operation and only affects status register.  P1OUT’s value is after;  **mov** operation: 00001111b  **bis** operation: 10101111b  **bic** operation: 10100000b  **bit** operation: 10101010b | | |