Name: George Landis

Lab: Lab\_04 I/O Using Logic Operations, and Bit Fields LEDs and Switches

Class: ECET 30903

Date: 1/31/19

Professor: Elizabeth Freije

1. Understand the Program’s Purpose
   1. State the problem/purpose:
      * + Writing a program that will produce a bit field without and within
   2. Then Identify
      * Inputs: List the hardware input(s) needed?
        + Switches
      * Process: Simples steps needed to preform program logic
        + Call one of the 2 Functions for Bit Fields
        + (Within those two functions do the same thing)
        + Check if each bit is high else go low
        + (Check every bit separately)
      * Outputs: List the hardware output(s) needed?
        + 8 LEDs
2. Design/Assign the hardware configuration. (For embedded system design)
   1. Create a table to identify and organize I/O hardware configuration assignments. Make sure to only include the I/Os the program needs.
      * + Table is below

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Identify I/Os hardware configuration assignments** | | | | |
| **I/Os Needed** | **Port name** | **Port address** | **purpose** | **Initial setting for 8-bit Port’s DDRx** |
| 8 dip switches | PORTB | 0x01 | input | DDRB = 0x00 default setting |
| 8 LEDs | PORTC | 0x04 | output | DDRC = 0xFF (sets bits 0-7)  Set DDRC memory mapped to 0x06 |

* + - What hardware I/Os are needed? Identify each I/O by hardware name.
      * 8-bit dip switches & 8-bit LEDs
    - Assign I/O hardware pin/port names and memory mapped Port address:
      * MC9S12XEP100 chip

8-bit Dip switches are controlled by PORTB at memory mapped address 0x01 8-bit LEDs are controlled by PORTC at memory mapped address 0x04

8-bit LEDs Initial setting by DDRC memory mapped address 0x06

* + - Set the I/O’s data directional register accordingly. Is the I/O used as an input or an output?
      * Dip switches are used as inputs: since all I/Os default as inputs you do not need to set the corresponding Data directional register.
      * LEDs are used as outputs: all 0-7 bits of PORTC are used to control the LEDs, so, set these bits as outputs….DDRC = 0xFF;

1. Design the Program Logic:

Break down each task that the program must perform into a series of logical steps.

* Check a bit-state on the dip switches (is the bit on or off?)
* Light the LEDs bit by bit
  1. Develop an algorithm pseudo code:
* Check each switch with bit field
* Turn on or off each corresponding LED for each bit.
  1. Flowchart the algorithm:

Flowchart: a diagram that graphically depicts the steps of an algorithm, where specific symbols represent each step in the algorithm.

A screenshot of a cell phone

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| **Identify function(s)** | | | |
| **Return data type** | **Function name** | **Parameter list** | **purpose** |
| void | Square | (void) | receives nothing, Calculates square of a  number, returns nothing |
| void | MyFunction | (unsigned char) | Receives and unsigned character, put your  process description here, returns nothing |

* Identify any function(s) if needed for tasks identified. Will the function receive any arguments, will the function return a value, what variable(s) will the function need to perform its task? Can use table as exampled at end of this document.
* Identify variable(s). Determine if the variables are to be global, local, uninitialized, initialized.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Identify Variable(s)** | | | | |
| **I/Os Needed** | **Port name** | **Port address** | **purpose** | **Initial setting for 8-bit Port’s DDRx** |
| 8-bit dip switches | PORTB | 0x01 | input | DDRB = 0x00 default setting |
| 8-bit LEDs | PORTC | 0x04 | output | DDRC = 0xFF |
| Stepper Motor | PORTD | 0x05 | output | DDRD |= 0x0F (sets bits 0-3) |

What are the processes? Define them.

* Check a bit-state on the dip switches (is the bit on or off?)
* Light the LEDs