General Purpose Input/Output

Lab 1

Readings:

Atmel SAMD20 DataSheet

- Section 21: PORT

Required Materials:

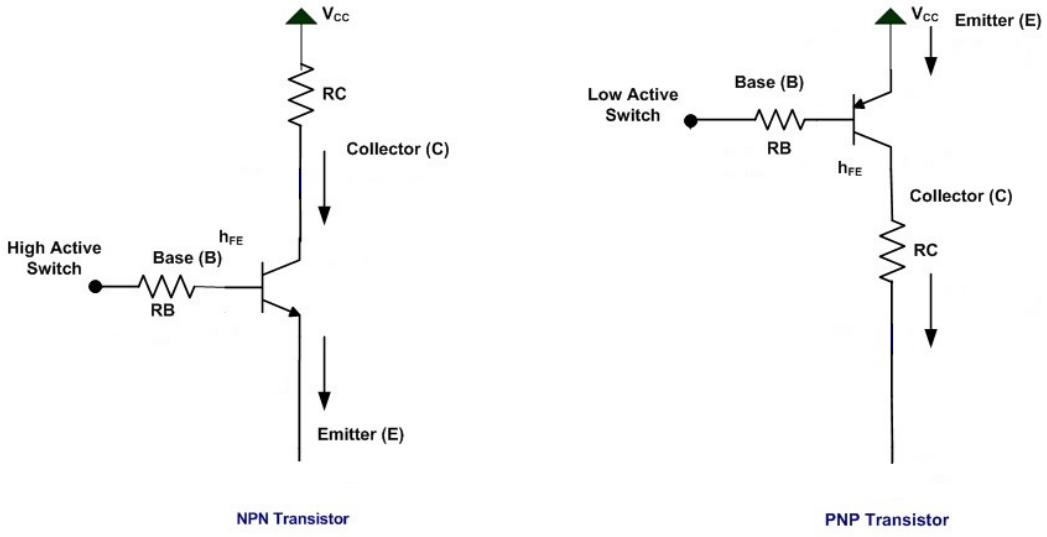
Atmel Sam D20 board

Lab Description:

Students will develop a calculator using the keypad and 7-segment display as the input/output devices. As a minimum, the calculator should be able to add and subtract two integers numbers (with sign). The digit key “#” will represent the 'add' (+), “*” will represent subtraction (-), "A" as the backspace, and "B" as equal (=). Do not worry about consecutive add and subtract operation as well as over/underflow (more than 9999 or less than -9999) condition. Extra credit will be given to project with additional function such as implementing multiple, divide, decimal point or floating point.

Theory of Operation and interfacing circuit:

1. Circuit (PNP transistors, LED current consideration, keypad matrix, pull up resistor)
    - The 7-segments display is activated by placing a high logic to the common anode. At the common anode of the 7-segments, there is a PNP transistor connected. The PNP transistor acts as a switch. When applying a low logic at the base of the transistor, voltage (Vcc) from the emitter pin is sent through to the collector pin, which will activate the LED.



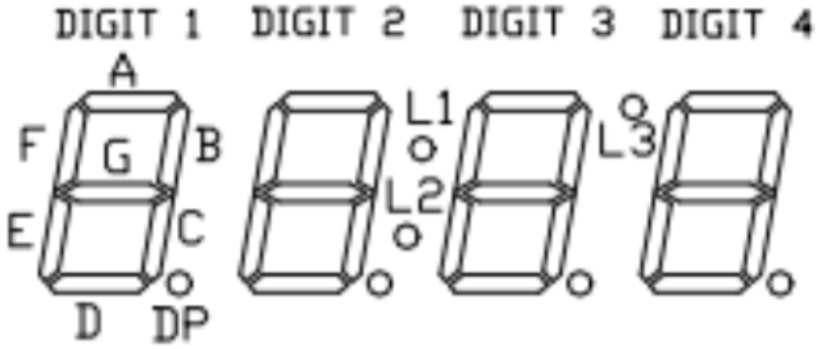
Bipolar Junction Transistor as the logical High Active (NPN) and Low Active (PNP) Switch

- If a 7-segment display is powered through the common anode, the LED’s will become active low. If it is powered by the common cathode, a high logic will then be required from the LED.
- In the keypad, there are 4 rows and 4 column pins. When you press a button, a specific row and column is connected together. Think about a button as a light switch. When you turn a

switch on, two wires are connected and turned off, the two wires are cut off. The same logic can be applied to the keypad.

2. Scanning (both display and keyboard)
    - In order for the keypad and display to work, the display (7-segment) must be multiplexed with the keypad. This means that the SAMD20 must constantly update/power the 7-segment and the keypad rows at a fast rate. Note that the base of the PNP transistors are connected to the “rows” of the keypad.

3. BCD to 7-segment decoding
    - Once an input is read from the keypad, that input needs to be displayed onto the 7-segment display. Pins PB00 to PB06 are connected to A to G respectively with DP connected to pin PB07. Digit 1 to Digit 4 are connected to pins PA04 to PA07 respectively.



4. Software debouncing (simple state machine)
    - Software debouncing is when a software detects and recognizes that only a single signal will be recognized upon a single opening or closing of contact (between wires or pins). Recognizing software debouncing is important because when a consumer presses a button on any type of keypad, they do not want to repeat a key press more than intended (when holding a key down).

Peripheral and Coding info:

```
Address-        SAMD20 Syntax Code
-------------------------------------------------------------------------
0x41004400   -      Port *por = PORT_INSTS;
offset 0x00  -      PortGroup *porA = &(por->Group[0]);
offset 0x80  -      PortGroup *porB = &(por->Group[1]);

offset 0x00  -      DIR.reg
offset 0x04  -      DIRCLR.reg
offset 0x08  -      DIRSET.reg

offset 0x10  -      OUT.reg
offset 0x14  -      OUTCLR.reg
offset 0x18  -      OUTSET.reg
```

offset 0x20 - IN.reg
offset 0x40 - PINCFG[x]
1u << xx - PORT_P(A/B)xx

C:\...\("project_name")\src\ASF\sam0\utils\cmsis\samd20\include\component\component_port.h

Required Task: ***please comment your code**

Task 1: (week 1)

- Turn on-board SAMD20 LED on and off (blinking) using only addresses and pointers. The on-board LED is on PORT_PA14.

Task 2: (week 2)

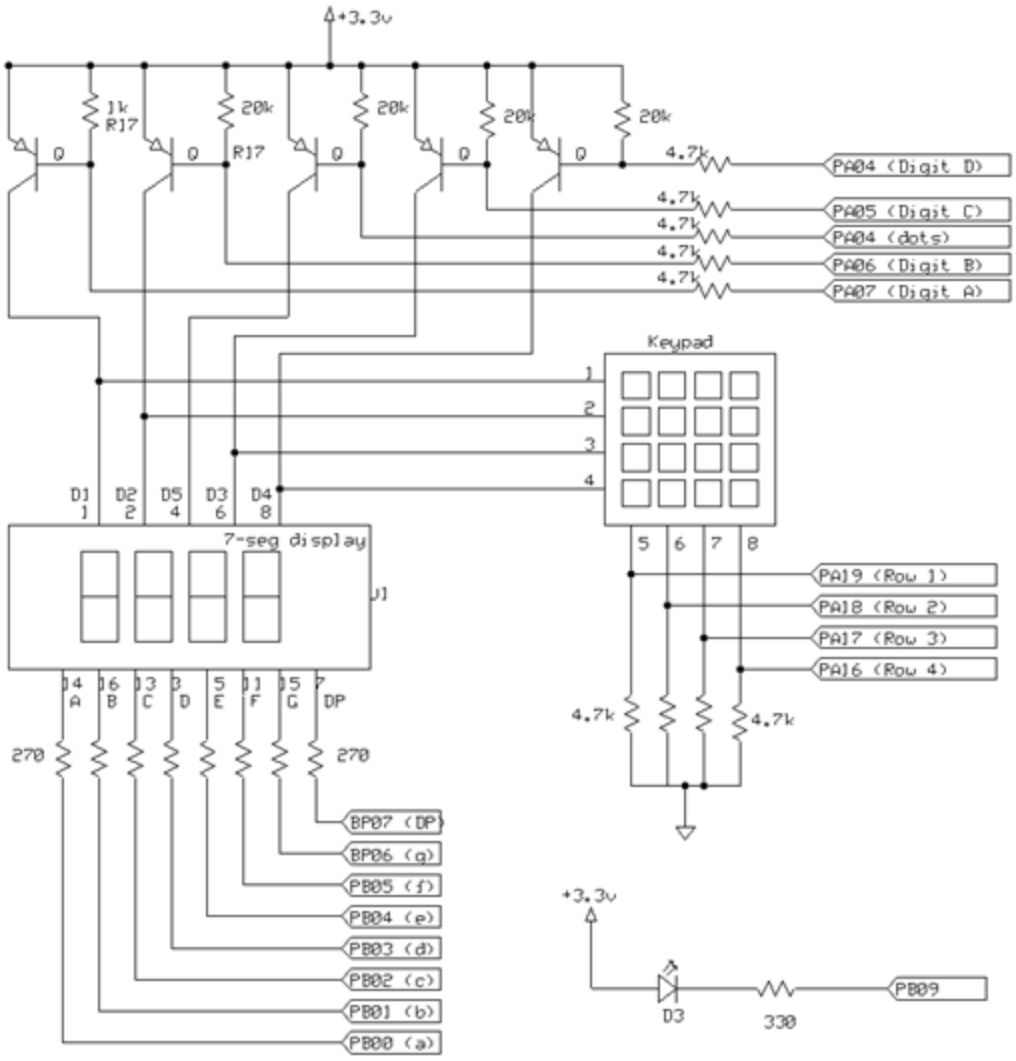
- Display 4 unique symbols onto the 7-segment display. *(ex: display "1 2 3 4" or "A, C, 5, F")*
  - Use the appropriate ports corresponding to the external board.
- Get the keypad to detect a response and display onto the 7-segment.
  - **Debouncing** - If someone was to hold a button down, make sure the software is not repeating the same key press and displaying it onto the display.
  - **State Machine Format** - Create "states" in the "main" section of the code. For example, one state could be used to check/read a keypad and another state will be used to check for debouncing. A third state could be used to do arithmetic functions or other extra features.

  hint: *When updating the 7-segment display at a fast rate, make sure that you are not updating the display "too fast".*

Task 3: (week 3)

- Create a program that can imitate the basic functions of a calculator. It should do the basic addition and subtraction. Do not worry about numbers larger than 7-segment display. Extra credit can be given out to those who do more than what is required.

GPIO Schematic:



References:

Atmel SAMD20 Datasheet

- Section 5: I/O Multiplexing and Considerations
- Section 8: Product Mapping (address locations)
- Section 13: Clock System

7-segment Display Data Sheet

http://media.digikey.com/pdf/Data%20Sheets/Lite-On%20PDFs/LTC-4627JR.pdf

Keypad Data Sheet

http://www.grayhill.com/assets/1/7/Keypads_96.pdf