

情報科学実験 B

第 1 回レポート

【担当教員】 石尾 隆 教員

【提出者】 内藤 圭吾 (09B14049)
ソフトウェア科学コース・2 年
u434666c@ics.osaka-u.ac.jp

【提出日】 2015 年 1 月 28 日

1 電卓の外部仕様

1.1 概要

受け付ける値域は、被演算数は2桁まで、演算数は1桁で、受け付ける演算は加算と減算である。

1.2 詳細

- 加算演算子は A、減算演算子は B を当てている
- 演算結果が3桁になる場合：正しく3桁表示
- 被演算数が0から入力された場合：演算結果がの頭に0が付加され表示される

2 エラー処理

エラーとなる入力が行われた場合、LEDを消灯して初期状態に戻る。エラーとなる入力は以下のものがある。

- 仕様を超えた桁数の入力
- 計算結果が負になる入力
- 複数の演算子の入力
- 数値を入力する以前の演算子の入力

3 動作例

入力	出力
9 A 2	11
10 B 2	8
98 A 3	101
A	エラー
1 A A	エラー
111	エラー
2 B 3	エラー

4 方針とその影響

外部仕様の決定における方針は、開発の簡易さである。実際エラー処理と、全て答えを表示し終わった後の初期状態に戻る処理は、同じ処理をしているので、共通のコードを利用している。また、入力を被演算数2桁以下 + 演算子 + 演算数 の形に限定することで、入力の終了タイミングを考える必要が無く、演算子の入力の後演算数が入力されたタイミングで計算の処理を行っている。

5 電卓内部の数値表現と演算方法

電卓内では数値は各桁ごとにスタックに格納されている。スタックは DS 命令で領域を確保して、x レジスタをスタックポインタとして利用している。また、演算子は別の領域 OPE に格納している。よって、”19 A 2”と入力された場合、入力直後とその演算後で、スタックは以下の様に変化する。

スタックの位置	格納されている値
2	2
1	9
0	1

表 1: 入力直後 sp = 3

スタックの位置	格納されている値
2	2
1	1
0	2

表 2: 演算後 sp = 2

演算方法は、数値が各桁ごとにスタックに格納されているので、演算数入力段階で、スタックの上 2 つの値が、演算数の 1 桁目の値と、演算数の値となっている。よって、1 桁目同士で計算を行い、その計算結果が 10 を超えるかを確認し、超えた場合はスタックのさらにひとつしたの値 (演算数の 10 の暗い) に 1 加える。もし、演算数の 1 桁目の格納されているのがスタックの底で、かつ桁上がりがあった場合は DS で確保した領域 CALI のフラグを立てる。結果を表示するときに、CALI フラグが立っていた場合は先にその値を出力することで、3 桁目の出力を行う。演算の過程における、スタックの変化は以下の用である。入力は先ほどと同様に”19 A 2”とする。

スタックの位置	格納されている値
2	2
1	9
0	1

表 3: 入力直後 sp = 3

スタックの位置	格納されている値
2	2
1	11
0	1

表 4: 一桁目同士の演算を実行 sp = 2

スタックの位置	格納されている値
2	2
1	1
0	2

表 5: 桁上がり処理実行後 $sp = 2$

6 他の方針との比較

今回は開発の容易さを方針としたので、演算方法は各桁ずつ演算を行ったが、演算の実装にはほかにも幾つかの候補があった。まずひとつは、入力された値を入力されるたびに 10 倍しながら足し合わせていき、10 進数の値にしスタックに格納する方法。この方法のメリット・デメリットとしては以下のものがあげられる。

- メリット
 - － 乗算・除算の実装が容易
 - － 入力の制限桁数の変更が容易
- デメリット
 - － 入力のたびに演算を行うので操作が煩雑
 - － 出力の際に、各桁毎に分解する必要がある

また、今回実装に利用した各桁での演算を行う方法のメリットデメリットとして以下のものがあげられる。

- メリット
 - － 最低限の機能の実装が容易
 - － 10 進数に変更しないので入出力が容易
- デメリット
 - － 桁数の制限を拡張するのが煩雑
 - － 乗算の場合桁上がりの数を確かめるために複数回減算を行う必要があるので煩雑
 - － 除算の実装が困難

以上の理由から、機能の拡張を行う場合は入力のたびに 10 倍していく方法の方がよいと考えられる。

7 回路図

7.1 課題 1

以下の図 7 は課題 1 の回路である。プッシュボタン 1 の出力を行うピン PB1 から、プログラムの入力を割り振った PTA0 へと接続する。しかし、プッシュボタンではチャタリングが生じるので、その間でチャタリング防止回路を挟んでいる。

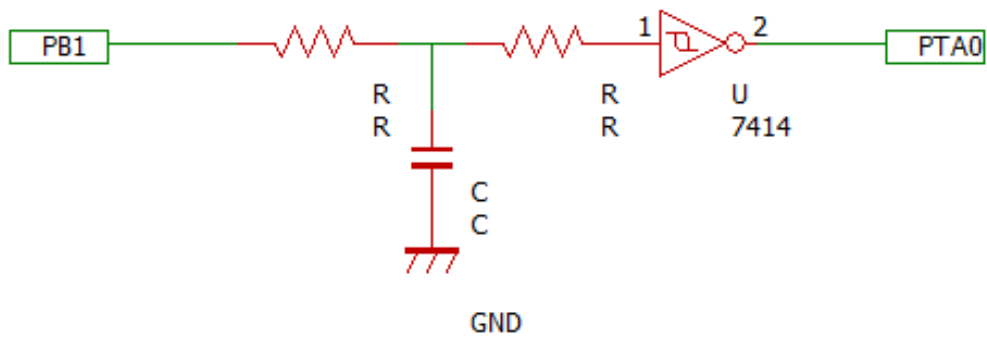


表 6: 課題 1 回路

7.2 課題 2

課題 2 の回路図は以下のようになる。

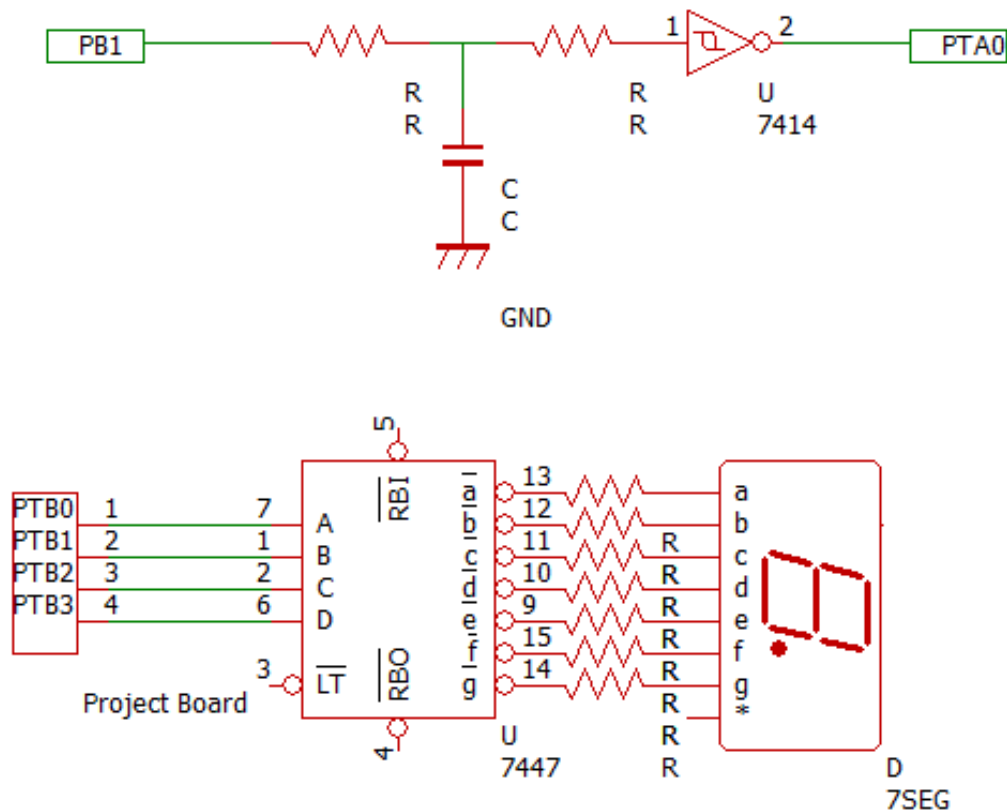


表 7: 課題 2 回路

課題 2 の回路は図 6 から分かるように 2 つの部分によって構成されている。各部の説明を行う。プッシュボタンの入力部は課題 1 と同じなので説明は割愛する。以下の図 8 は課題 2 の出力部である。課題 2 では次に出力する値を 4bit の 2 進数で出力しているので、それを IC7447 で 10 進数に変換し、その結果を 7segLED に出力している。

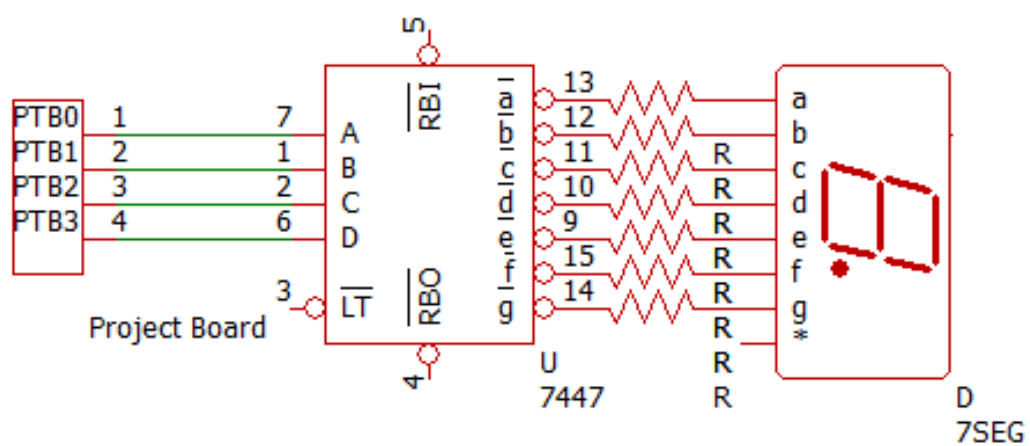


表 8: 課題 2 回路出力部

7.3 課題 3

課題 3 の回路図は以下のようになっている。

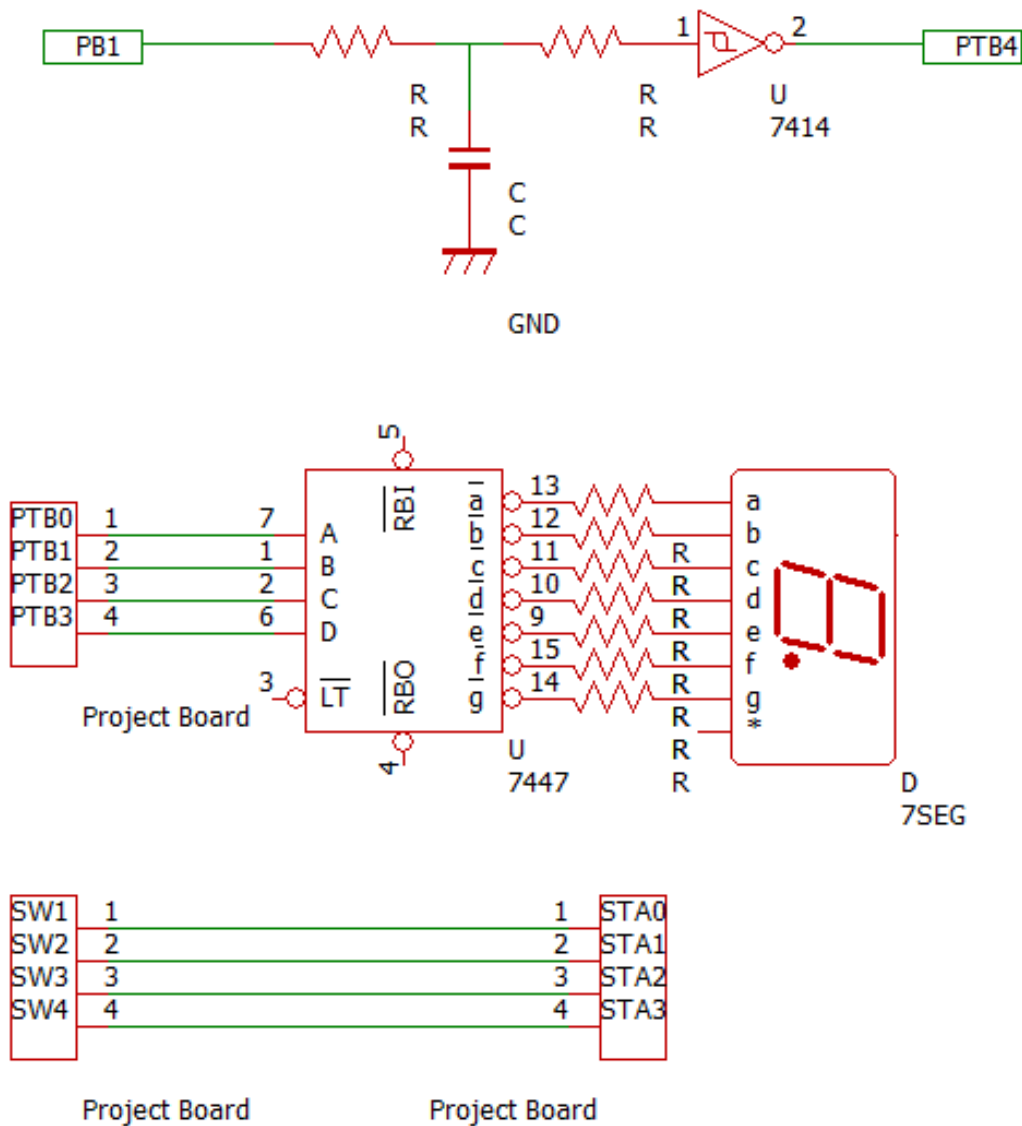


表 9: 課題 3 回路図

課題 3 のにおいても、プッシュボタンの入力及び出力部分は課題 2 と同じなので、説明は割愛する。以下の図 10 は数値の入力部である。今回数値の入力にはディップスイッチを利用したので、ディップスイッチの出力ピンである SW1 1 ~ SW 1 4 を、それぞれプログラムで数値入力として設定した PTA0~3 に接続した。

SW1	1	1	STA0
SW2	2	2	STA1
SW3	3	3	STA2
SW4	4	4	STA3

Project Board

Project Board

表 10: 課題 3 数値入力部

7.4 課題 4 以降

課題 4 以降は課題 3 の回路と同じものを利用したので割愛する。

8 他授業との関連

今回の 7segLED を利用した値の出力は、情報科学実験 A で用いたものと同じ構造を利用した。また、スタックを用いた演算は、情報科学基礎での CASL を用いた電卓の実装で利用したものを使った。