Scientific Calculator

Basic objectives

A) Make sure the btn are divided into 6 by 6 btn
B) Use slide switches for other operation

1) Make sure to be able to display decimal numbers.
2) Display -ve decimals.
3) Then, make sure the calculator could do power & square root.
4) Then, make sure the calculator can do sine, cosine, tangent
5) Make sure the calculator could perform complex numbers
6) Make sure the calculator can perform repeated operations

For number shifting: num1 = 65250 (65535 - 285) [To shift 8 bits]
num2 = 285
num2 = num2 >> 8

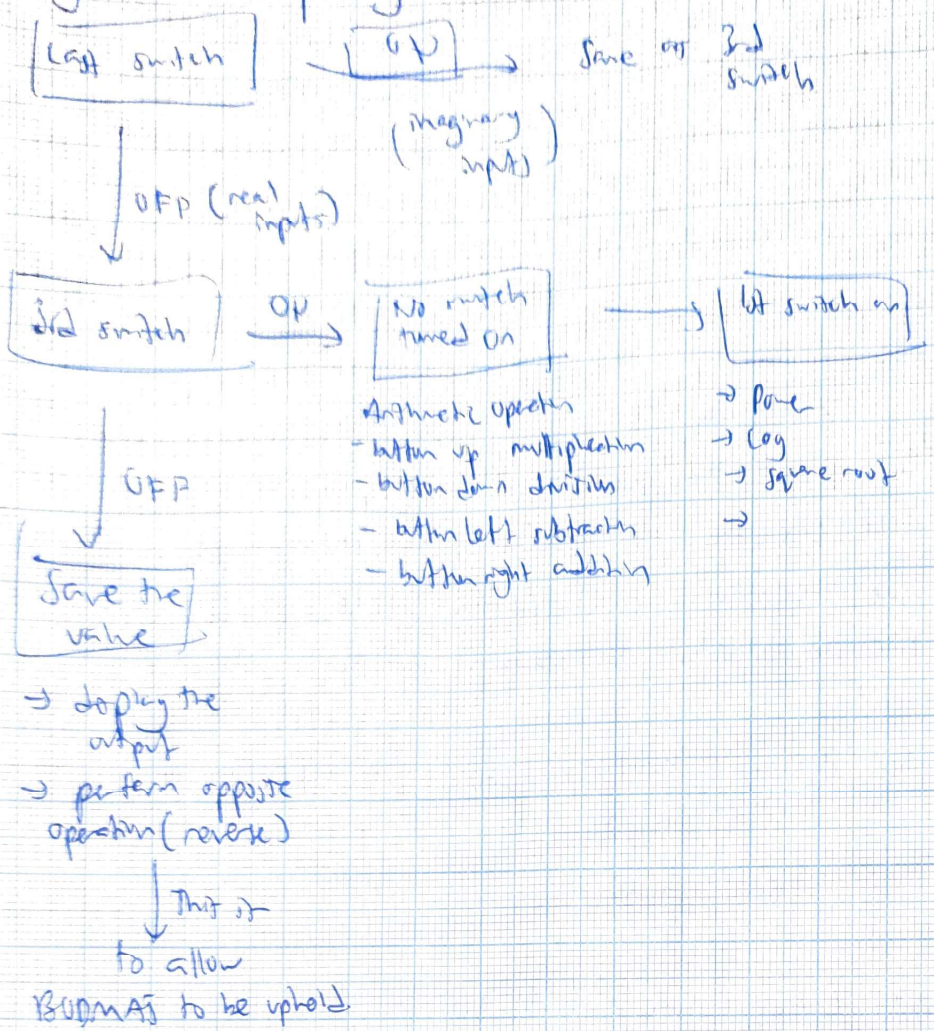
[To shift 6 bits] num1 = 4095 - 63
num2 = 63
num2 = num2 >> 6
→ Use Two's Complement to make sure negative number inputs are possible.

For the repeated operation, num1 + num2 = ans
ans + num2 = ans
ans - num - num → maybe declare a global variable to be saved.

6)
[Input by input.]
→ Can be sorted using pointers & void function
Test: → The void function does not have an integer to cast (Error)
solution → Add '&' at the ~~pointer~~ variable that will be called at the pointer's position.

→ The calculation() function must add more cases in the switch statement because more operations will be performed.

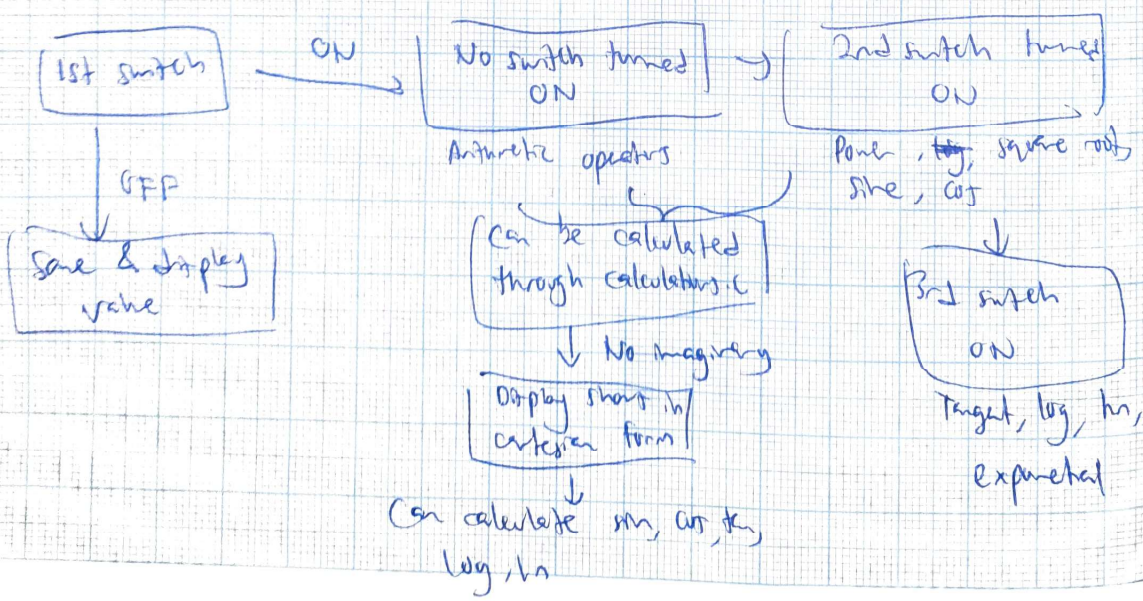
3) By flow-chart planning:



Test:- The flowchart is not stable as inputs will be needed to ~~input~~ input separately, not one time

- If-else statements would need to be performed more, thus making it not modular.

* New flowchart is needed.



switch1 = slideSwitchIN && 0x1000 (SW12)
switch2 = slideSwitchIn && 0x2000 (SW13)
switch3 = slideSwitchIn && 0x4000 (SW14)

} Assign the value for the switches

If switch 1:
- L - minus
- R - add
- U - multiply
- D - divide

switch 2:
- L - power
- R - square root
- U - sine
- D - cosine

switch 3:
- L - tanget
- R - expweka
- U - nat_log
- D - logarithm

↑ Buttons

- The variable must be in complex
- <complex.h> library is imported & included in calculations.c for operations to work.
- <complex.h> have cexp, cpow, ccos, csin, ctan, clog.
- cannot compute log base 10 (need to find a way to display)

For display purposes, creal & cimag of output can be taken by two complex variables inside main.c function.

- The values can be compared using if-else statements to make sure that whole number, decimal or negative decimal is shown.
- A variable must detect which numbers type are the image and real values of the result.
- decide variable is used, decide = 1 (decimal)
  = 2 (whole number)
  = 3 (negative)
  = 4 (negative decimal)

Test:
- The negative decimal & positive decimal of '0.' and below cannot be displayed
- seg7_display.c has been modified, still same result.

Problem: the variable inside the calculations() function of the operation does not detect decimal for 0.1 - 0.99.

(58) Fixed: - A new double complex variable is introduced into the function to then be multiplied by 10.
- creal & cimag will be assigned to an signed integer (s16) deci_I1, deci_R1, to detect the remainder
- remainder detect by (deci_I1 % 10 != 0)
- Works!

Calculate() function for trigonometric functions

- sine(), cosine(), tangent() functions are created
- Uses cCos(), cSin() and cTan().
- Inputs must be in degrees.

Test: The answer of sin(90) by FPGA board is 0.8. Input is still not in radians

Solution: - added PI definition, which the is used by val = PI/180.
- the code becomes *_z = cSin(arg * val).

-Still does not work.

Solution: - another variable a = arg * val is used,
- code works since only 1 variable can be inside cSin(a).

Need to add codes for square root power, log & ln

- cpow() can be used, it works
- csqrt() also works
- clog() works for natural logarithm.
- log() does not work on complex numbers, need to find a way to implement log base 10 or just input real numbers to make it function.

Used law of logarithm $\log_a b = \frac{\log_n b}{\log_n a}$,

used clog(arg*_z)/clog(10) for log base 10 to work.

Testing for repeated calculations, and all the operations

Switch 1 → ((3+7)÷6)×(1+2) = ? ✓
→ ((2÷3)−7)×1 = ? ✓
→ −8÷3÷1 = ? ✓

Switch 2 → power → (2+1) ✓
→ sq-root → $\sqrt{(13 \div 2)}$ ✓
→ sine → sin(−0.3) ✓
→ cosine → cos(2^2) ✓

Switch 3 → tangent → tan($\sqrt{-2}$) // tan(3+7×5) ✓
→ exp → $e^{\sin(2\times3)}$ ✓
→ nat-log → ln(1−3) ✓
→ logatm → log(sin 4) ✓