Lab 1-2 FPGA

reason 4 hex b/c first 4 digits of a bin # ⟹ 1 digit of a hex #
and no amounts of bin digits corresponds to the 1st digit of a base 10 #

Lab 2.1

22) errors: when displaying 3 ⟹ ⅎ ⟶ change 0110001 to 0110000
when displaying 8 ⟹ ⊟ ⟶ change 0000001 to 0000000
when displaying E ⟹ Ⅎ ⟶ change 0010110 to 0000110

Lab 2.3

⟹ 6 5 4 3 2 1 0
X X X X X X X

33) when the output > 15 or F a 1 is carried over to the next digits place ie F + 1 = 10

34) when the sum is > 15 (or F) cout is high on the last adder in series this high runs through a not gate turning into a low which stops the 7-segment LED for the number in the second digits place from being constantly cleared by its blank input this allows the LED to display the hex # 1 in the second digits place because MF constant is always inputting "0001" this allows carry over for #'s > F so that if we do F+1 = 10 as opposed to the thing breaking and displaying F+1 = 0

Lab 2.4

34) logic elements used 42, we have used < 1% of the FPGA
total # of logic elements on FPGA: 114,480

Lab 2.5

Decimal divider works up to 99 b/c $(100)^2$ = 5 digit # and we don't have enough LED displays to display a 5 digit decimal #

base 10 # ⟹ to the max 8 bit # is 255 ⟹ this requires a 3 digit display and 2 dividers b/c 1 divider will give us the 10's place digit & the second will give us the 100's place, since we don't have a 3 digit input display and only a 2 we will use 1 divider to give us base 10 #'s up until we get a 3rd digit ie 99 after ward the second digit becomes a hex digit (it always was though b/c we didn't run it through a 2nd divider)