Gareth Peters (260678626)
Peter Quinn (260689207)

October 13, 2017
ECSE 323

Lab 1: Modulo 13

Gareth Peters (260678626)
Peter Quinn (260689207)

g07_Modulo_13

Description

This circuit is designed to take a 6 bit positive unsigned integer as input. It outputs two positive unsigned integers. The floor13 output is 3 bits wide and is number of times 13 fits into the input. The Amod13 output is 4 bits wide and represents the remainder after dividing the input by 13.

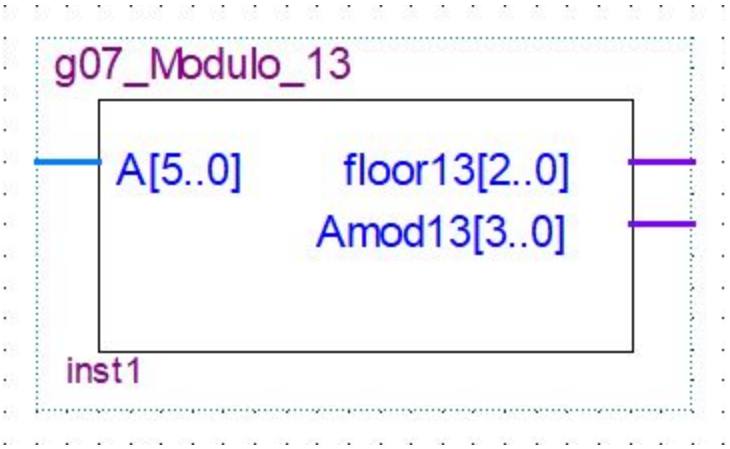

Fig. 1: Symbol of g07_Modulo_13 circuit

This circuit was implemented using the following algorithm taken from the Lab 1 notes provided on MyCourses [1]. This circuit uses five 8 bit adders (see g07_adder) and one NOT gate to achieve the design requirements.

1. Floor13 is calculated by dividing the input by 13. In order to reduce the complexity of the circuitry involved, this is accurately approximated by multiplying by 5/64.
    a. The input is shifted left twice, and then added to itself. This multiplies by 5.
    b. The sum is then shifted right 6 times (taking into account the possible carryout from the adder). This divides by 64 and produces the 3 bit output floor13
2. Floor13 is multiplied by 13, to find the largest multiple of 13 that fits into the input
    a. Floor13 is shifted left 3 times, and then added to floor13 shifted left 2 times
    b. This sum is then added to floor13
    c. This final sum is 13*floor13
3. Two's complement is used to subtract 13*floor13 from the input
    a. 13*floor13 is inverted using a NOT gate that acts on the whole bus line, then 1 is added to this inverted number. This creates the twos complement of 13*floor13
    b. The input and the inverted number+1 are added together
4. Amod13 is obtained by taking the 4 least significant bits of this sum

Gareth Peters (260678626)
Peter Quinn (260689207)

October 13, 2017
ECSE 323

Schematic

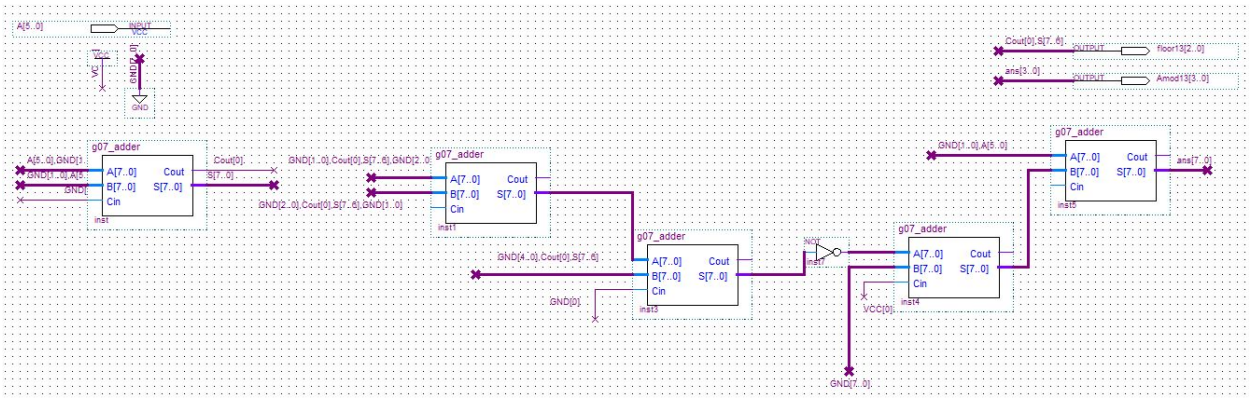

Fig. 2: Gate level diagram of g07_Modulo_13 circuit

Testing and Simulation

Given the formula A=floor13(A)*13+Mod13(A), we notice that each respective output in Fig. 3 satisfies this formula. An observation can also be made of the output for Amod13, which is that none of the outputs are greater than or equal to 13. The pattern for Amod13 is to increment for every incrementation of the input A, and reset back to zero once the input reaches a multiple of 13. These two observations follows the properties of the Modulo operator.

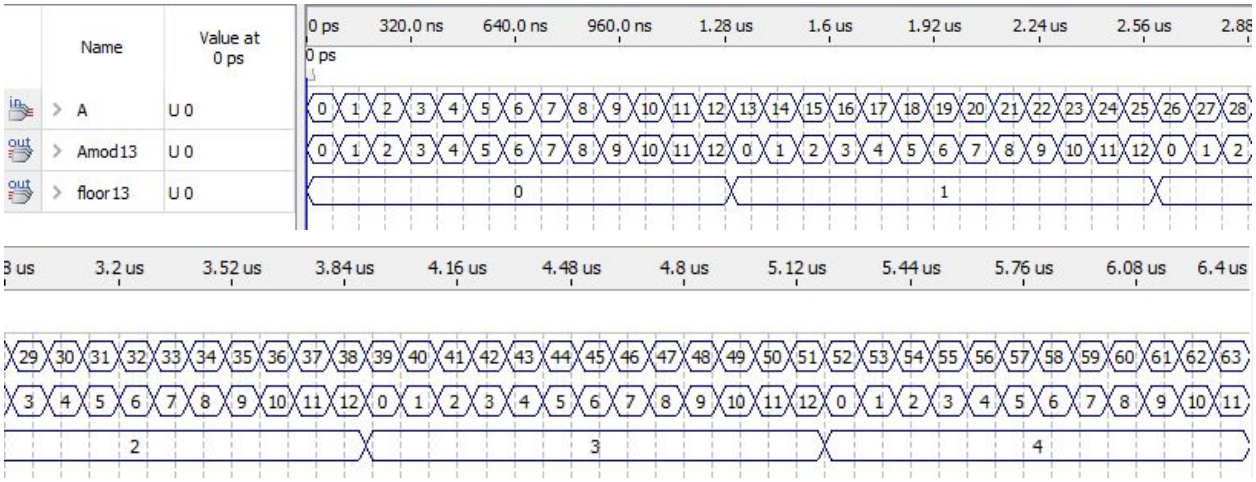

Fig. 3: Waveform Simulation of the g07_Modulo_13 circuit including floor13

g07_adder

Description

This circuit is an 8 bit adder. It takes two 8 bit inputs, A and B, and a 1 bit carry in Cin. It outputs the 8 bit sum of A, B and Cin, as well as the 1 bit carry out Cout. The circuit is implemented using eight 1 bit full adders chained together in a ripple carry configuration [2].

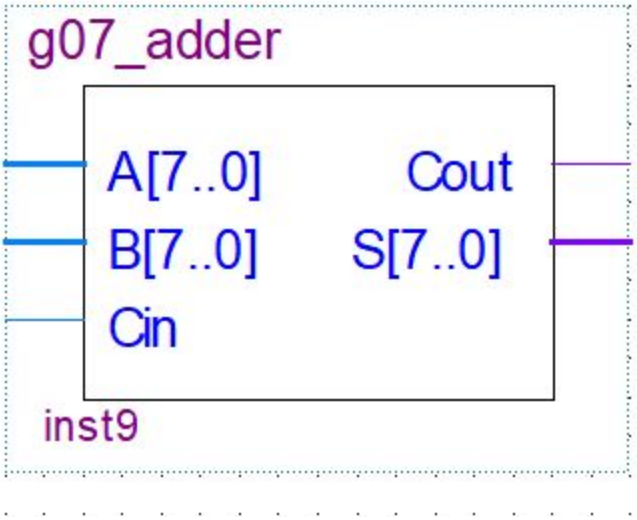

Fig: 4: Symbol of g07_adder (8 bit adder)

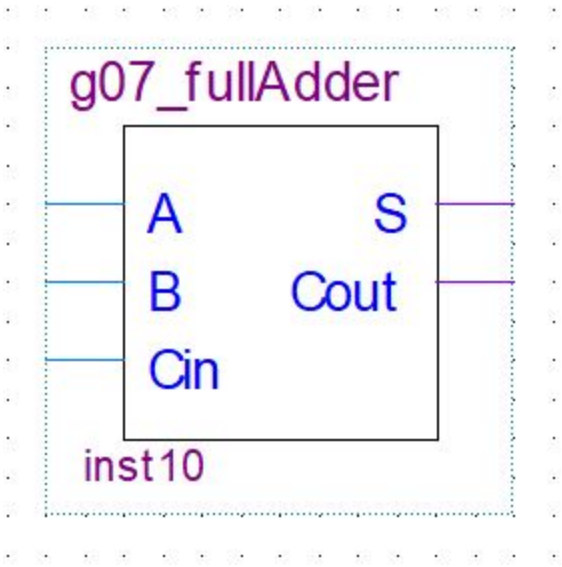

Fig. 5: Symbol of g07_fullAdder

Schematic

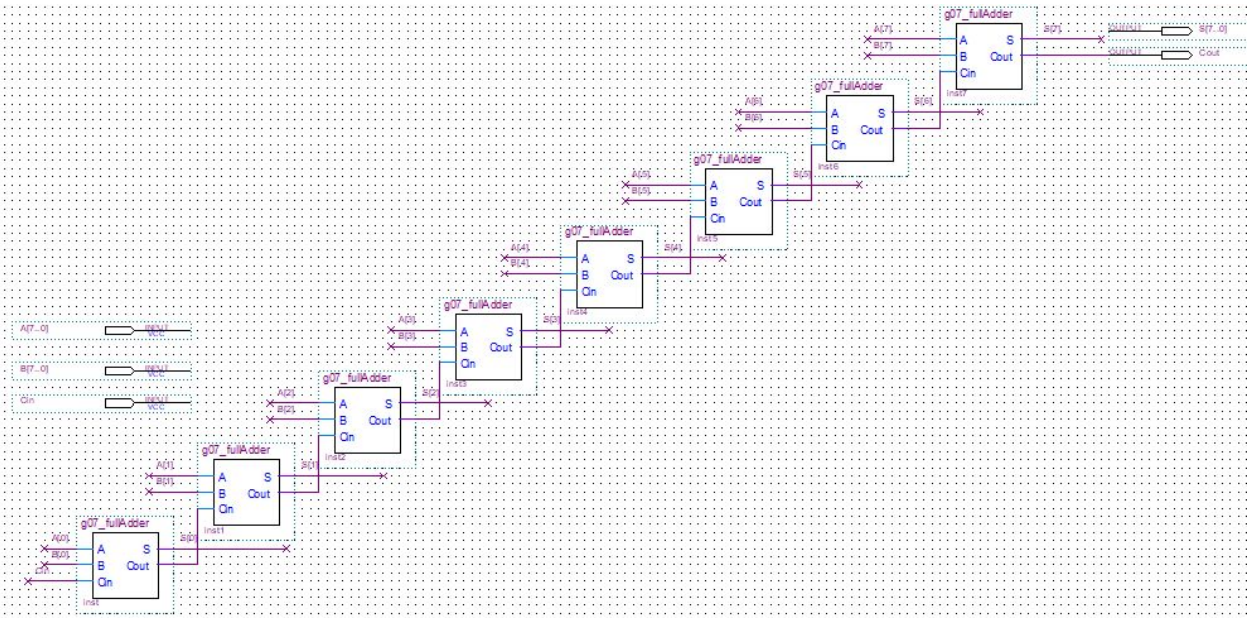

Fig. 6: Gate level diagram of g07_adder (8 bit adder)

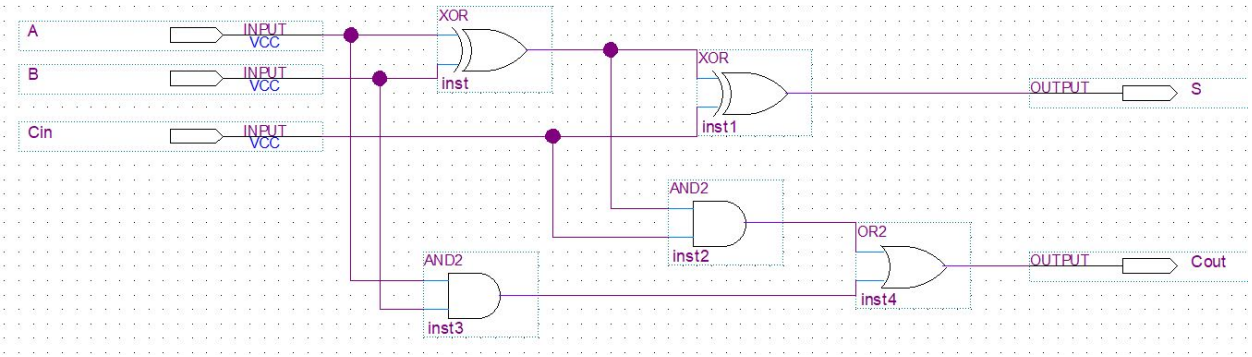

Fig. 7: Gate level diagram of g07_fullAdder

Testing and Simulation

Given the simulations from Fig. 8, it is observed that the S represents the summation of A and B and Cin. If A+B+Cin is greater than 255, an overflow occurs and the most significant bit is lost. The Cout output provides the signal for the most significant bit, and we can observe that Cout concatenated with S (in binary) represents the true summation of A and B.

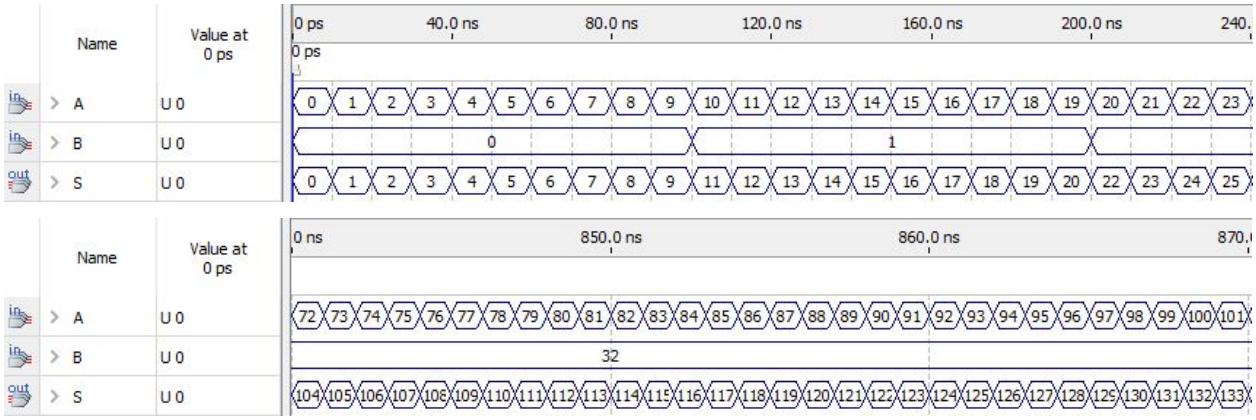

Fig. 8: Waveform Simulation of g07_adder, showing for a sample of values that A+B=S for S<256

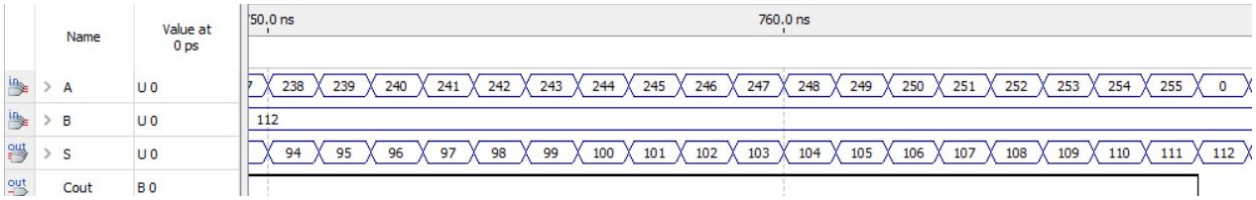

Fig. 9: Waveform Simulation of g07_adder, showing that when S exceeds 255, Cout is activated

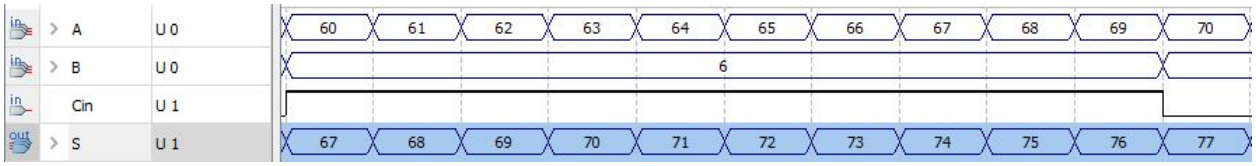

Fig.10: Waveform Simulation of g07_adder, showing that adding Cin gives A+B+Cin=S

g07_comp7

Description

This circuit is designed to take two 7 bit inputs A and B and compares the two to see if they are equal. The output AeqB is 1 bit. It is 1 if A = B and is 0 otherwise. This is done by comparing each bit of A and B separately. The circuit uses seven two input XOR gates for each bit of A and B and three 3 input AND gates to check if the bits of A and B are equal.

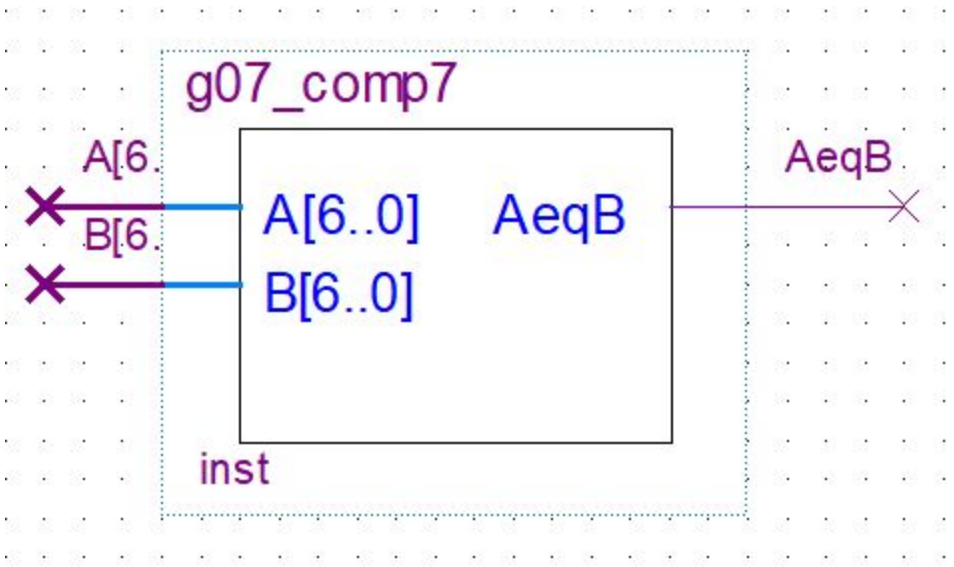

Fig. 11: Symbol of g07_comp7

Schematic

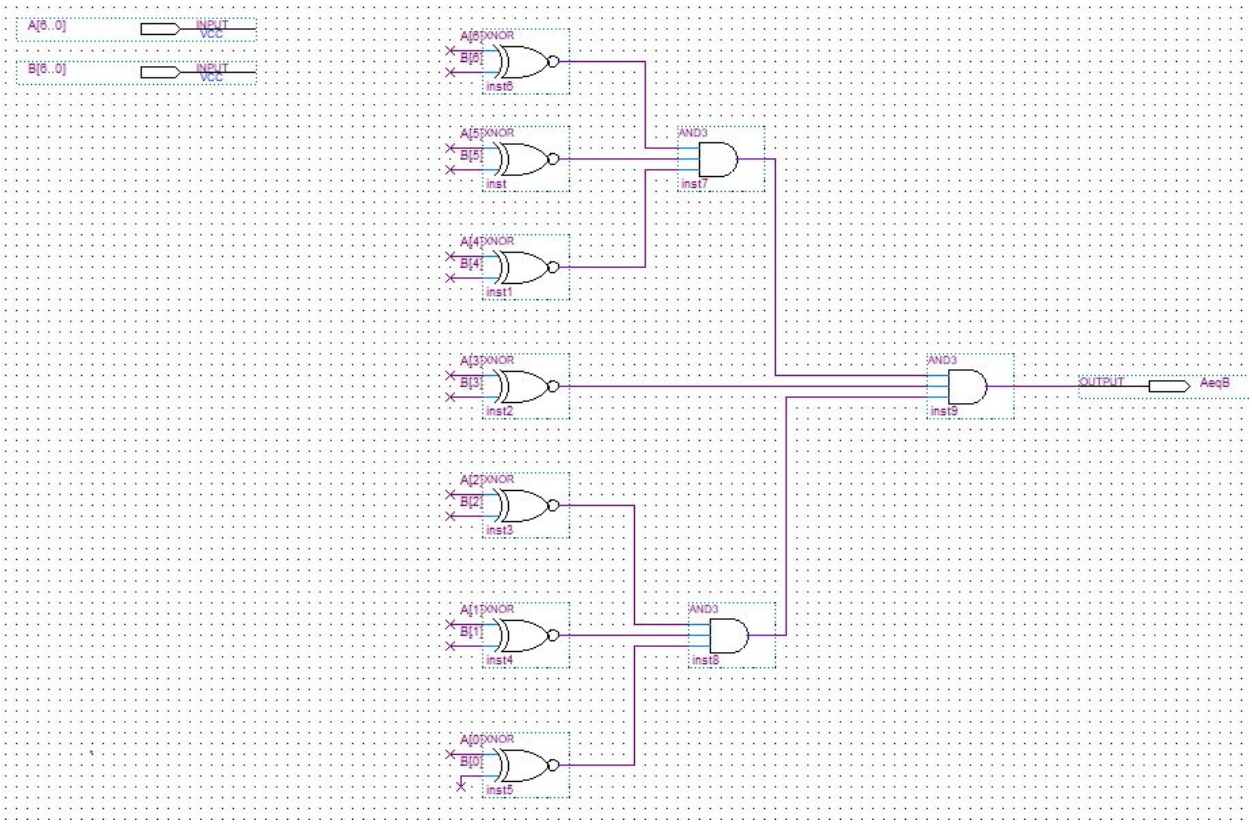

Fig:12: Gate level diagram of g07_comp7

Testing and Simulation

AeqB is always 1 when A = B and 0 otherwise. The input A was simulated such as to increment 128 times before B was incremented.

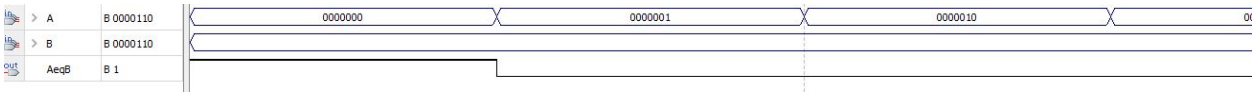

Fig. 13: Waveform simulation of g07_comp1, showing that the output AeqB is 1 when A=B (note that B is 0 in the whole waveform) and 0 otherwise.

References

[1] J. Clark, Lab #1–Using the Altera Quartus II Software. Montreal: McGill, 2017.
[2] "Adder (electronics)", En.wikipedia.org, 2017. [Online]. Available: https://en.wikipedia.org/wiki/Adder_(electronics). [Accessed: 12- Oct- 2017].