Detailed description
We choose to do the ALU.
An arithmetic logic unit (ALU) is a combinational digital electronic circuit that performs arithmetic and bitwise operations on integer binary numbers.
The inputs to an ALU are the data to be operated on, called operands, and a code indicating the operation to be performed.
We have 3 operands : A and B are four bits each, and Cin is one bit.
Our ALU can do four different operations, so we need a two bit code to know which operation to do. These input are s0 and s1.
The operations are :
00 Unsigned add
01 Unsigned subtraction
10 Test for bit equality
11 Divide by Two

We made our project by realizing 6 schematics.
The main schematic is the ALU itself.
It contains: -2 ripple carry adder.
-A decoder
-A module to compare bits
- 4 enabler

Each ripple carry adder is made using 4 full adder .
A full adder is a logical circuit that performs an addition operation on three one-bit binary numbers. The full adder produces a sum of the three inputs and carry value.

The add function is done by using a simple carry adder.
The subtraction function uses the complement method. We invert B and use the ripple carry adder to do A+Band we add one. We add one using the Cin input of the rripple carry adder module.
The Test for bit equality has its own schematic, which is just a 2 Level circuit. We use a XNOR gates for each element of A to compare it with his B homologue. Then we use a 4AND gate to see if A completely equal B.
The Divide by Two function hasn't got any module, it is a simple shift left of the input. It goes directly in the enabler, the entry are just shifted.

The ALU calculates everything simultaneously. Then we use an enabler to select the result. Each function has its own enabler and All the enablers are the same. An enabler just passes the inputs into outputs when the enable line is on.

To turn on the right enable line according to s0 and s1, we use a decoder.

Conclusion :

Our ALU works according to the specified certification.
We made a gate level system taking care to do as few levels as possible.
In addition to work, our solution is optimized.
However, you have to pay attention to some details:
- It is not planned to be able to subtract a large number from a small one. The displayed result is not valid.

- We can divide only a binary number of 6 digits. Additional digits will be ignored.

There are many ways to improve our system.
-On the circuit, we did not know how to access the positive wire.When it was necessary to add 1 during the complement method for the subtraction, we used a stratagem thanks to an inverter. This is certainly not optimized.
- We can check which is the largest number for subtraction, or turn off the LEDs if A <B.
- The divide by 2 function could divide larger numbers.
- We used the 25mhz clock. Our system could be 4 times faster if we use the 100mhz clock.
- On this FPGA, we can use the four 7-segment LED display to show our result instead of LED.
-The point of an ALU is to be multifunctional. But to add functions, we need more input and output on our FPGA. Or we can use the serial ports to add many switches and LED. So we can add almost as many functions as we want.