

YOUR Group Details:

Group No:	47
Group Members (Student IDs):	Dhiren Harindu Purasinghe (20604846) Zhang Shuming (20705228) Jiang Yuze (20593889) Ahmed Mohamed Eltayeb Mohamed (20712132)

Justification of YOUR Circuitry Diagram Design:

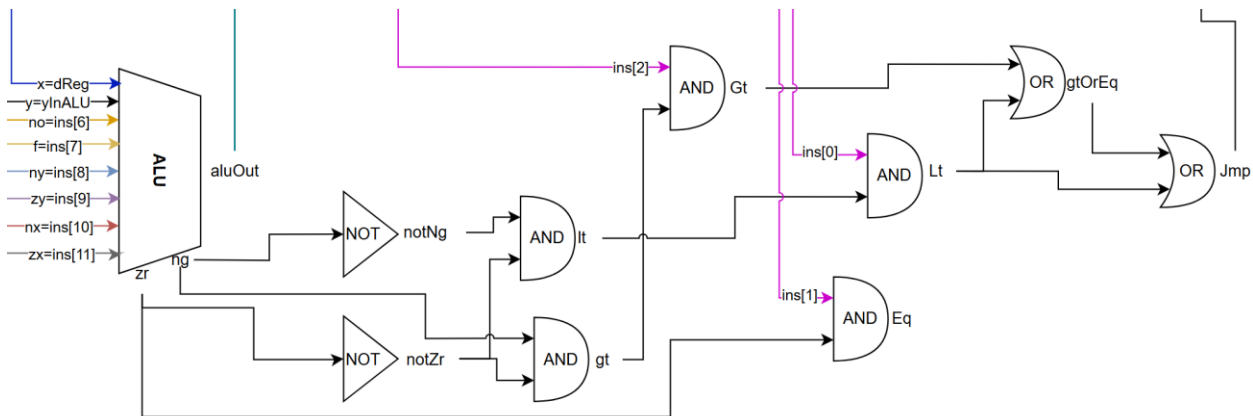


Figure 1. improved jumping condition model

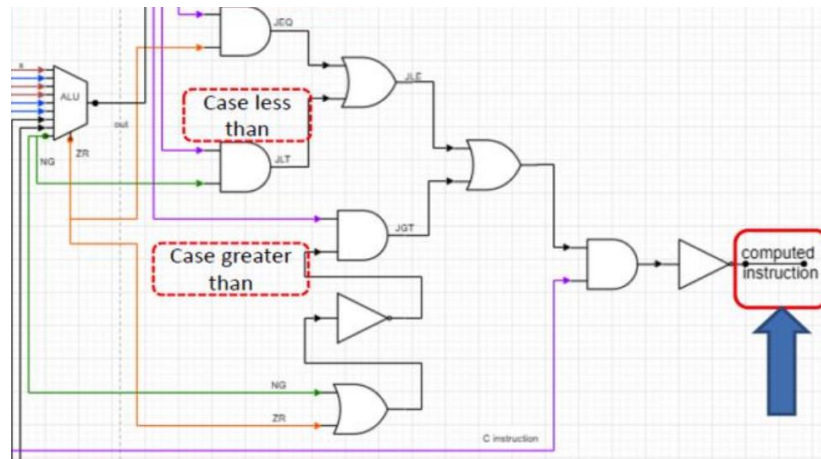


Figure 2. original jumping condition model

In coursework 1, our group creatively designed a CPU. This CPU follows the skeleton of the original version provided, with major changes to the jumping model. This allows the same program to run with much higher efficiency (Table 1).

There are three main changes implemented to the jumping condition model:

1. The new design takes fewer steps for the NG flag to reach the greater than (GT) condition. Unlike the original design, the NG flag must undergo an OR gate and a NOT gate to reach the same result. This helps the program to skip the pointless steps and get a more straight forward answer.

2. At the final step of the original model, the instruction must be reversed to get the final answer, which is unnecessary. Our design implements the NOT gates at the beginning of the diagram and integrates them into later calculations for simpler calculations.
3. The JLE (jump if less or equal) condition is redundant, since JLE checks if the result is zero or negative, and JGT (jump if greater than) checks if the result is positive, indicating that JGT is the negation of JLE. In our code, this is solved by simply using a condition of greater or equal to.

Compilation speed(s)	Original model	New model	Time difference
Slow	130.25	120.61	9.64
Medium	50.94	46.04	4.90
Fast	14.66	7.82	6.84

Table 1. comparison of the compilation speed for the two designs

Table 1 compares the compilation speeds of two models (Original and New) across three different compilation speed levels: slow, medium, and fast. The table shows that the new model outperforms the original model in each category. For example, at the Slow level, the original model takes 130.25 seconds, while the new model takes 120.61 seconds, resulting in a time difference of 9.64 seconds. Similar trends are seen at the Medium and Fast levels, where the new model consistently has shorter compilation times.