2. APRIL 2018

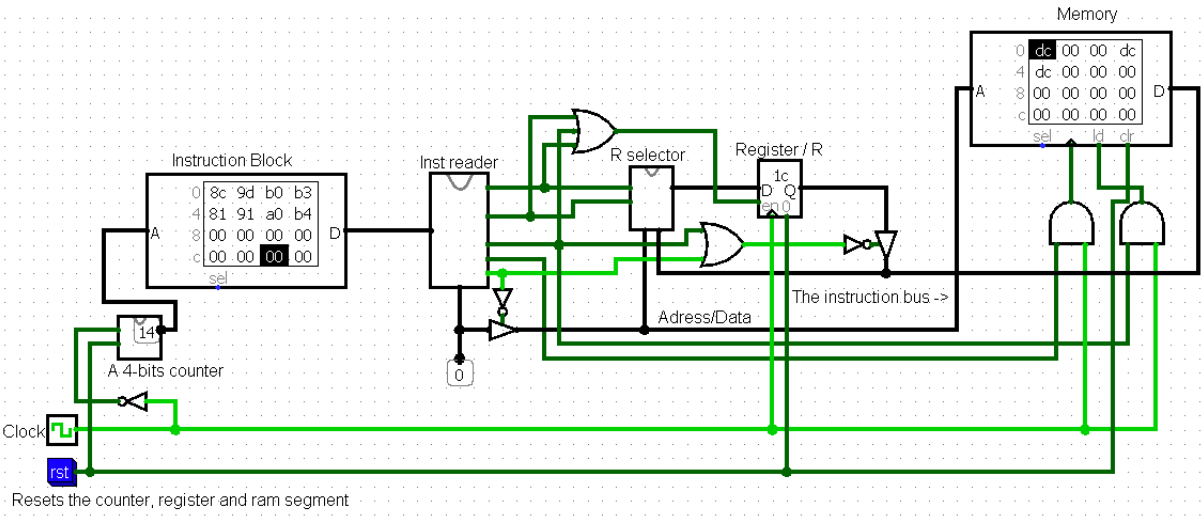
INF2270 – OBLIGG 2

«EN MINI-CPU»

JAKOBSKR

The CPU:

The CPU with Program 2 loaded:



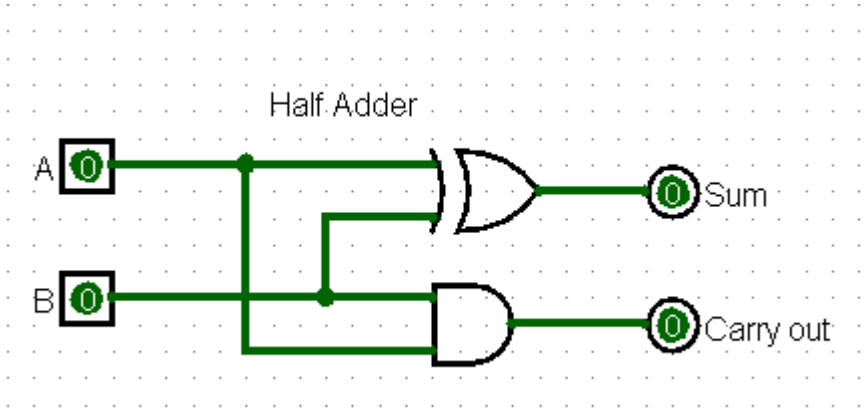
The ram is rising edge, while the register and counter is falling edge. I chose to implement it this way due to having some issues with the first and last instructions not being read.

The way this CPU works is that it reads the instruction block, splits the 8bit string into 2 4bit string where the MSB is the operation, and the LSB is the DATA/ADRESS. And then it executes the given operation on/with the given DATA. Either Writing into R, Reading R from Memory or Writing to memory from R. Then the counter increases and it reads the next instruction, until it finishes.

And rst resets the counter, the Memory, and the Register.

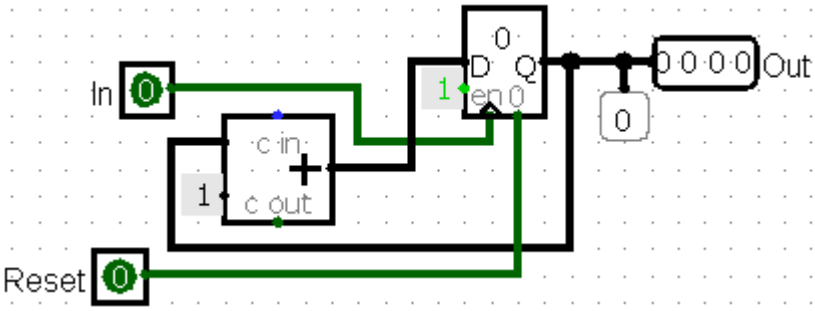
The Components of the CPU:

Half-Adder



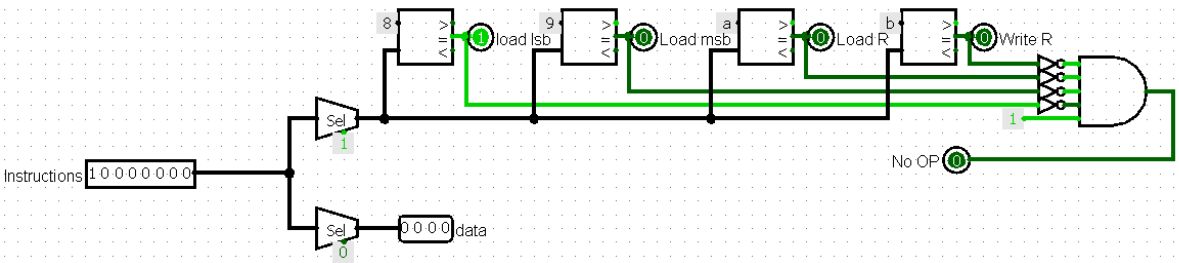
Just an ordinary half-adder

Counter



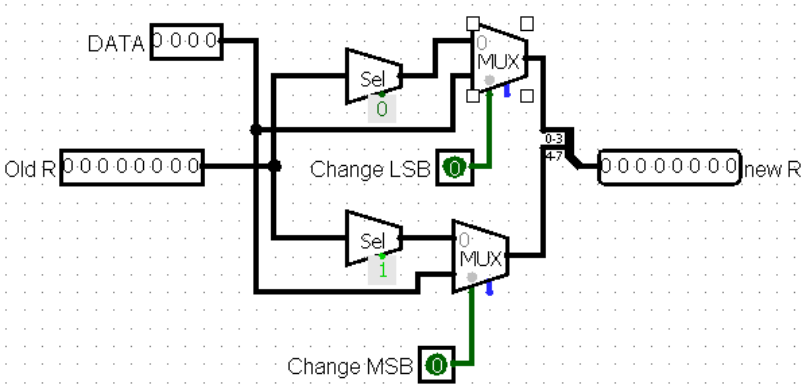
A counter using a 4bits adder and a 4bits register, when in goes high, the stored value in the register increases by one. When reset is high it resets the register. Output is the output.

Instruction reader



The instruction reader splits the 8bit string read from the instruction block, into 2 4bits strings where the LSB is Data and the MSB is the operations. It then checks if the operation is a valid operation or not. Outputs operation and data.

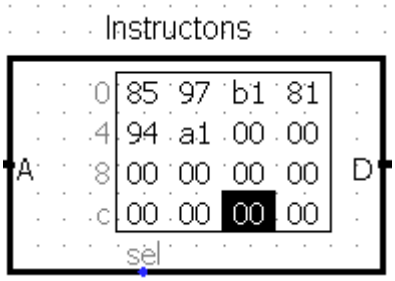
R Selector:



The R Selector splits the incoming R into its 4 LSB and 4 MSB. And then sends them to a mux, if "Change MSB" is active then R's MSB is changed to DATA. And vice versa if "Change LSB" is active then R's LSB is changed to DATA. If neither are active then Old R is the output, which happens when we read from the memory. There shouldn't be a situation where both Change MSB and Change LSB is active.

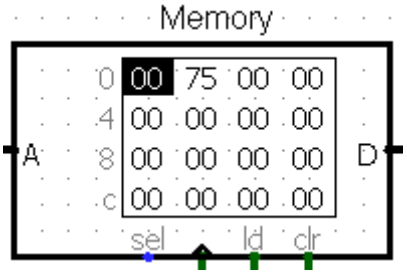
Execution of programs:

Program 1:

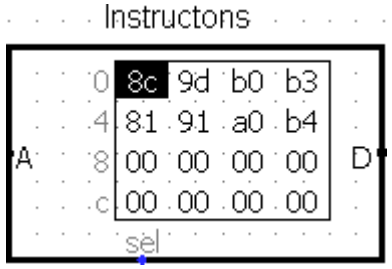


Expected result: 0 75

Result:



Program 2:



Expected result: dc 0 0 dc dc

Result: