**Hardware Implementation of Softmax Function**

Budget : 110$

Duration : If possible, in a week. If not by the 13th of January.

Project details:

Introduction

I'm working on a university project where I have to investigate the optimal amount of parallelization of the exponentiation block to accelerate the softmax layer. The softmax function is represented by:

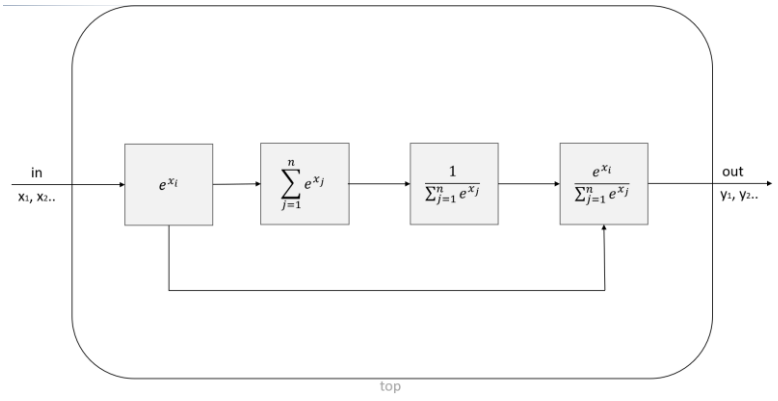
$$s\left(x_i\right) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}$$

My workflow is to :

Implement single cycle circuit of the function ➞ Pipeline it ( I'm stuck here )➞Parallelize the exponentiation modules ➞Evaluate the Area, Power and Latency/Delay.

The tools that I'm using are Quartus Prime Lite Edition of synthesis, and ModelSim as the simulator. Currently, I have separated the calculation into four main modules which uses Intel Quartus IP blocks. So, designing these modules from scratch is not required. They are :

1. Exponentiation module
2. Accumulator module
3. Reciprocal module
4. Multiplier module

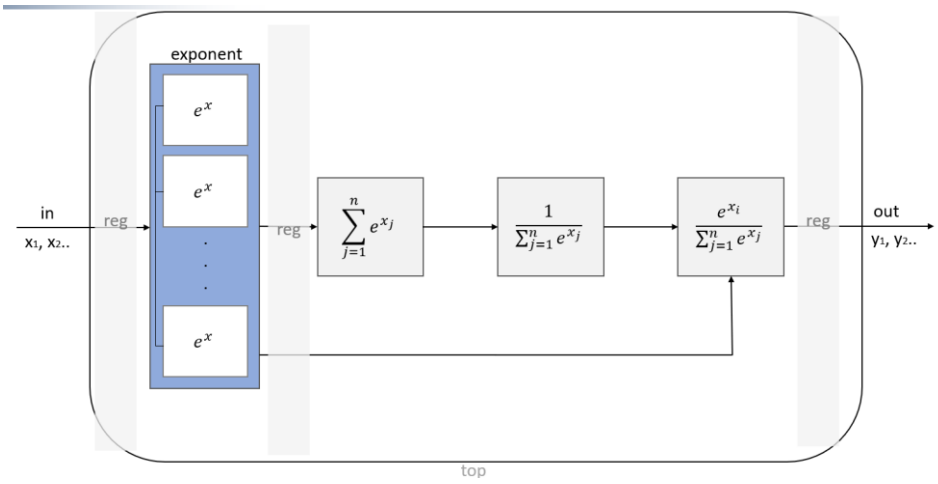



Picture 1. Baseline circuit

In this baseline circuit, input $x$ will first be converted to $e^{x_j}$, and then accumulated. The reciprocal of the accumulation will then be calculated and then finally multiplied with the $e^{x_j}$ that was calculated earlier. I have successfully tested this single cycle baseline circuit.

However, my final circuit is something to the likes of Picture 2. To increase throughput, I proposed 3 pipeline registers ( using RAM: 1-Port Quartus IP ). The input buffer, exponential buffer and the output buffer. Since each modules have its latencies, I am having trouble timing the writing/reading of the pipeline registers. Currently, I am solving the timing problem for the exponentiation stage. I

have made an address generator for the input buffer, **but I don't know how to make it so that the subsequent pipeline registers write/reads the correct address at the right time.**




Picture 2. Final circuit w/ parallel exponent & pipeline registers

What I need help with

I need help with pipelining and parallelization. If possible, evaluation will also be helpful but not required if the tools for it isn't available. In addition to the source code, I need documentation ( workings and testbench ) for each steps for reference & debugging.

My ideas for the pipelining is :

1) A control unit to control the timing of pipeline registers (?)
2) A unit that reads the address accessed by the registers for debugging

Other than that, if there are any additional modules needed to make the circuit work, please make them.

Specifications

Board : Cyclone V, any.

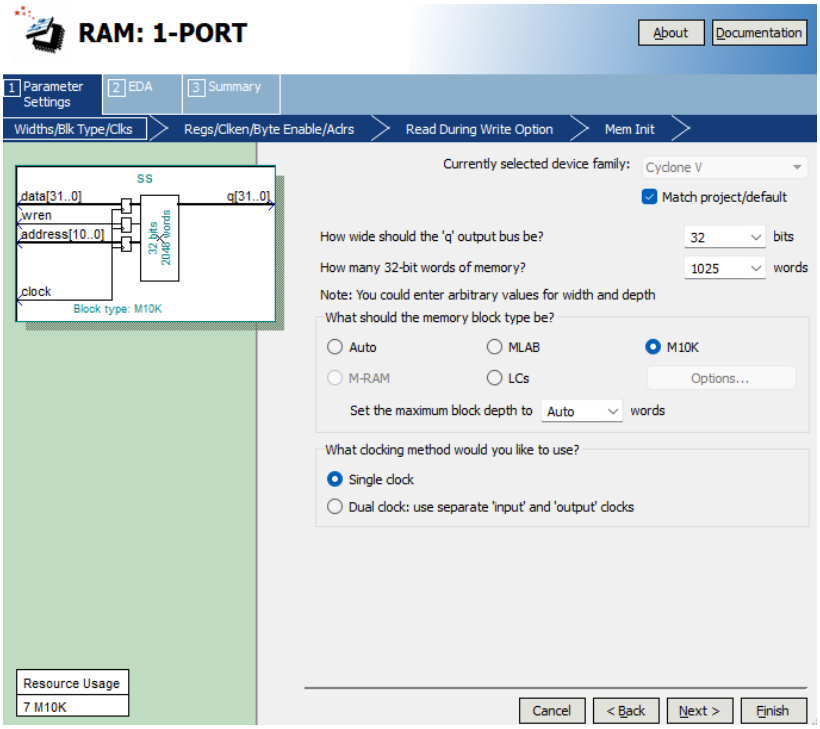
Target frequency: 200 MHz.

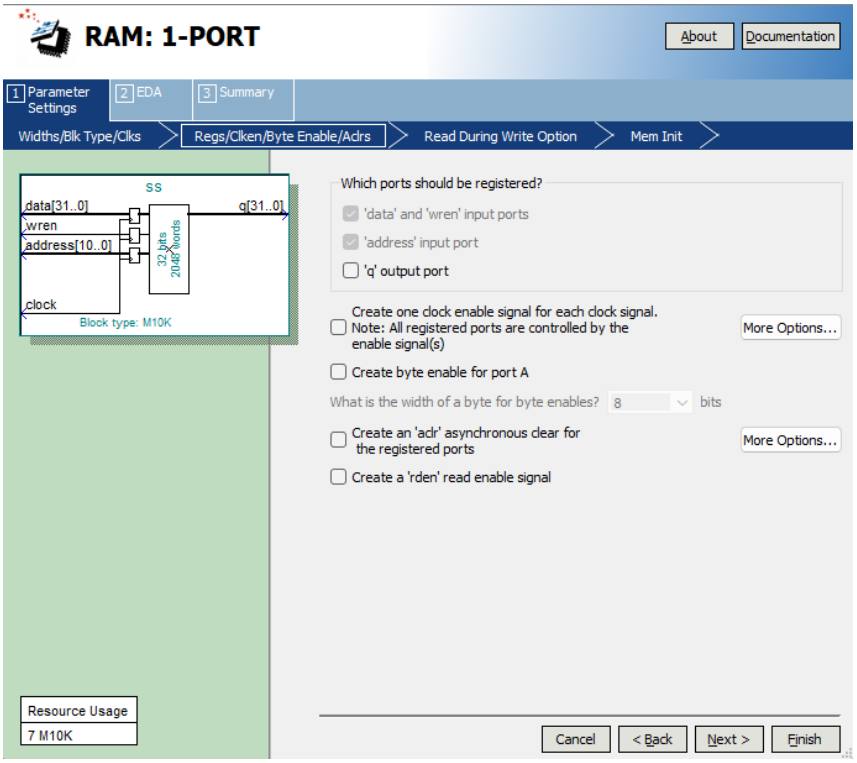
Input are 32-bit width single precision floating-point format.

The IP modules used are (please adhere to this):

| Modules | IP Name | Frequency ( MHz) | Latency (cycles) |
|---|---|---|---|
| Exponentiation | FP_FUNCTIONS Exponent | 200 | 23 |
| Accumulator *ignore the output signals outside of r. | FP_ACC_CUSTOM Default settings | 200 | 11 |
| Reciprocal | FP_FUNCTIONS Reciprocal | 200 | 15 |
| Multiplier | FP_FUNCTIONS Multiply | 200 | 7 |

For the pipeline registers, I'm using RAM: 1-PORT with these parameters




However, I don't know if the parameters or even the RAM I'm using is optimal. Feel free to change anything except for the bits and words.