Experiment 4

Accelerator and Wrappers

Mohammad Mahdi Abdolhosseini
810198434

Homayoun Mohseni
810198464

Abstract— **This experiment focuses on the integration of hardware accelerators in a System on Chip (SoC). SoC is an integrated circuit that combines various components into a single chip, including a processor, memory, I/O ports, and accelerators. Accelerators are specialized units designed to execute specific tasks efficiently at high frequencies. In contrast to CPUs, which handle multiple operations within fixed time intervals, accelerators offer higher operational speeds due to their simpler datapaths. By offloading tasks to accelerators, the overall speed of the SoC can be increased. The experiment explores the concept of handshaking in SoCs and the principles of accelerators. It involves implementing an exponential accelerator on an FPGA, which allows the CPU to delegate tasks, perform parallel operations, and access results stored in memory. Detailed topics covered include the Exponential Accelerator Engine, Exponential Accelerator Wrapper, and the implementation of the accelerator on an FPGA.**

1. Exponential Engine

This module takes a 16-bit input "x" and generates a 16-bit output "Fractionalpart" along with a 2-bit "Integerpart,". The accelerator operates when a complete pulse is received on the "start" signal and sends a "done" signal to the processor upon completion. To ensure design accuracy, a Modelsim simulation is performed with a testbench containing various input values for "x". Then we synthesized The design using Quartus II Software.

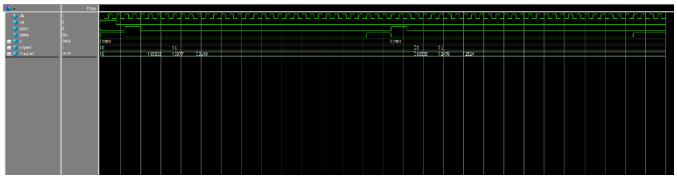

Fig.1 exponential engine simulation

Flow Summary	
Flow Status	Successful - Mon Jun 12 18:56:18 2023
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition
Revision Name	exponential
Top-level Entity Name	exponential
Family	Cyclone II
Device	EP2C20F484C7
Timing Models	Final
Total logic elements	101 / 18,752 (< 1 %)
Total combinational functions	100 / 18,752 (< 1 %)
Dedicated logic registers	60 / 18,752 (< 1 %)
Total registers	60
Total pins	38 / 315 (12 %)
Total virtual pins	0
Total memory bits	0 / 239,616 (0 %)
Embedded Multiplier 9-bit elements	2 / 52 (4 %)
Total PLLs	0 / 4 (0 %)

Fig.2 exponential engine synthesis report

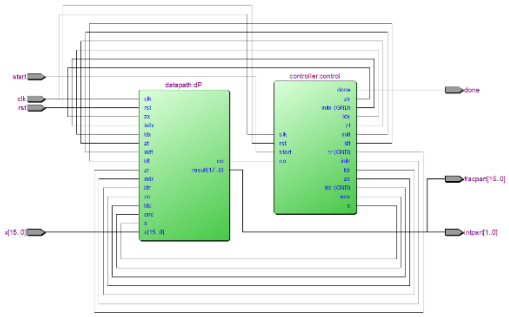

Fig.3 exponential engine high level diagram

Slow Model Fmax Summary

	Fmax	Restricted Fmax	Clock Name	Note
1	117.14 MHz	117.14 MHz	clk	

Fig.4 exponential engine maximum frequency report

The resulted maximum frequency is 117.14 MHz.

2. Exponential Accelerator Wrapper

The accelerator wrapper addresses the issue of timing overhead between the high-frequency accelerator and the low-frequency processor. By utilizing the free time, the accelerator can compute multiple exponential values, such as those required for the activation function in Deep Neural Networks. The wrapper implements a mathematical transformation that splits each input value into an integer and fractional part. The exponential function is then simplified using a shifter that shifts the fractional part by the integer value. The wrapper also includes tasks such as data loading, control signal generation, shift register operation, and FIFO storage. The controller manages the sequencing of calculations and data flow, while the FIFO stores the computed exponential values for later retrieval by the CPU. The design involves developing the state diagram and Verilog description for the controller, implementing the wrapper module, creating a testbench, and verifying the design with various input values, and finally, synthesizing the design.

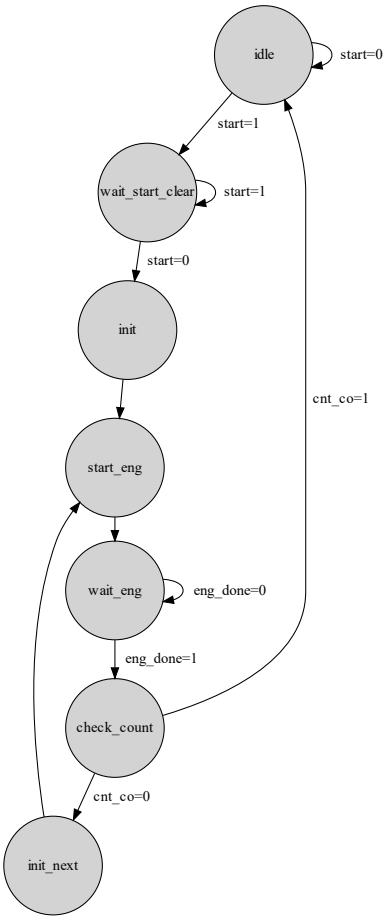

Fig.5 controller state diagram

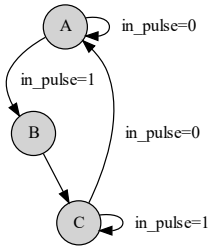

Fig.6 one_pulser state diagram

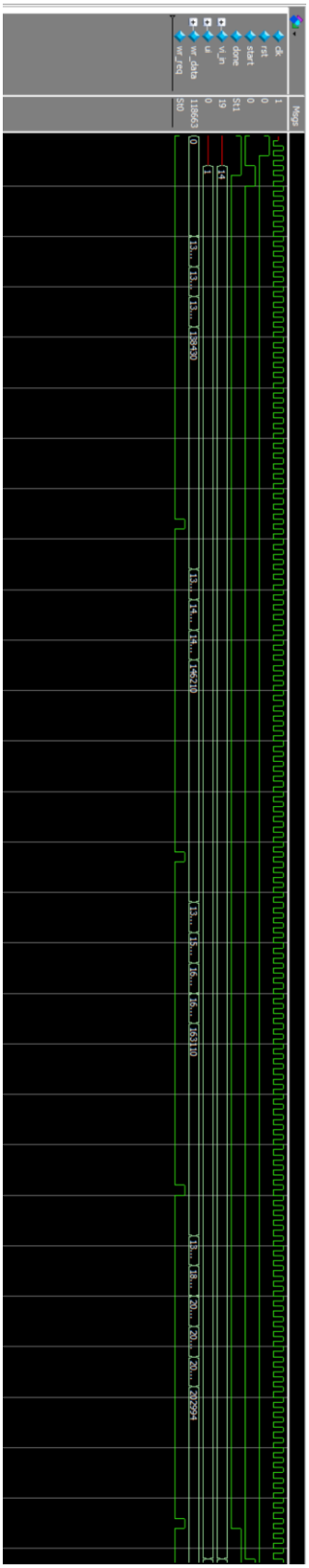

Fig.7 exponential accelerator simulation: batch 1

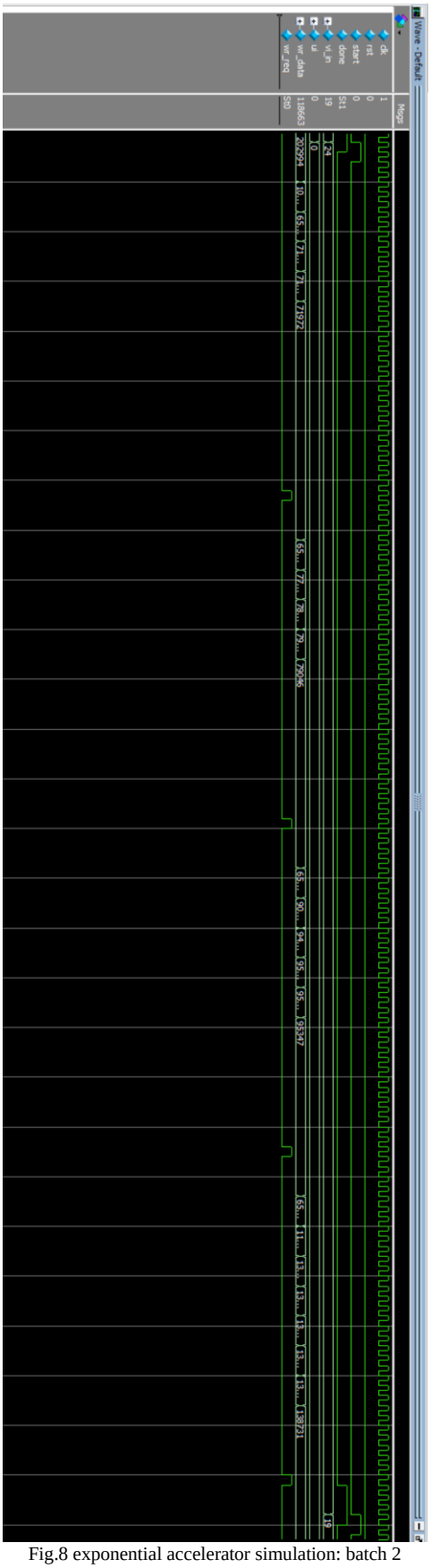

Fig.8 exponential accelerator simulation: batch 2

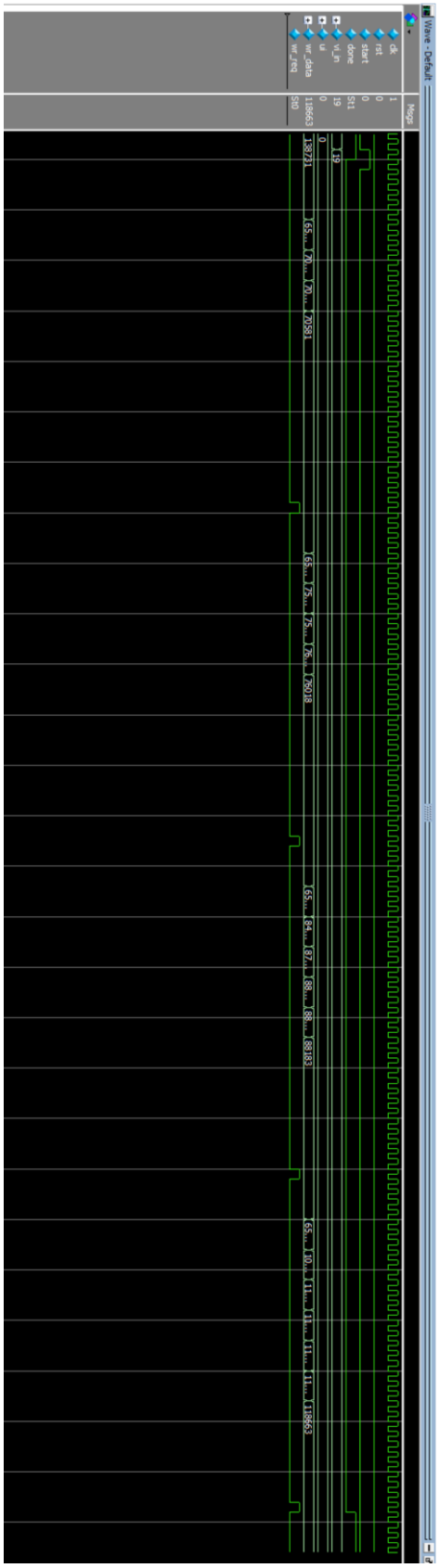

Fig.9 exponential accelerator simulation: batch 3

Below, we present the input and output values of the simulation.

batch 1:
inputs:
vi_in = 14
ui = 1
outputs:
wr_data: 138430, 146210, 163110, 202994

batch 2:
inputs:
vi_in = 24
ui = 0
outputs:
wr_data: 71972, 79046, 95347, 138731

batch 3:
inputs:
vi_in = 19
ui = 0
outputs:
wr_data: 70581, 76018, 88183, 118663

We have verified that the achieved values precisely match the expected values.

Slow Model Fmax Summary

	Fmax	Restricted Fmax	Clock Name	Note
1	114.76 MHz	114.76 MHz	clk	

Fig.10 exponential accelerator maximum frequency report

The resulted maximum frequency is 114.76 MHz.

3. IMPLEMENTING ACCELERATOR ON FPGA

In this phase, the wrapper design is synthesized and implemented on the FPGA. The synthesis report is included in the documentation. The wrapper's "done" signal is connected to LED[9], which will turn on after each round of estimating four values. The "start" pin of the wrapper is connected to SW[9] on the board, and it needs to be initiated at the beginning of each round. The fractional values (vi) are fed to the design using switches SW[8] to SW[3]. The most significant three bits of the fractional part are always set to zero. SW[2] and SW[1] are used for the integer part (ui), and SW[0] is used for the reset signal. The clock frequency of the wrapper is set to the 50 MHz clock of the FPGA. A one-pulser circuit, designed in a previous experiment, is used to generate the readreq signal for the FIFO. LEDs on the board are used to display the 21-bit result values, with one LED for "done," five LEDs for the integer part, and the remaining LEDs for the most significant bits of the fractional part. The design has been tested and delivered to the TAs.

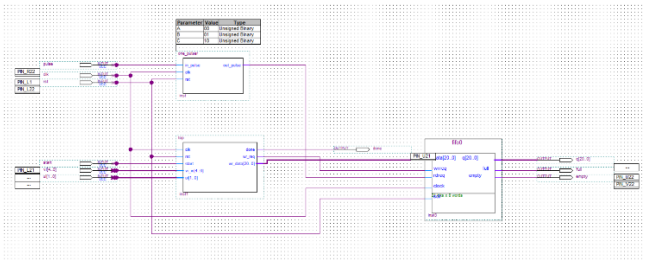

Fig.11 top design diagram

Flow Summary

Flow Status	Successful - Mon Jun 12 20:03:12 2023
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition
Revision Name	expnential_accelerator
Top-level Entity Name	expnential_accelerator
Family	Cyclone II
Device	EP2C20F484C7
Timing Models	Final
Total logic elements	171 / 18,752 (< 1 %)
Total combinational functions	169 / 18,752 (< 1 %)
Dedicated logic registers	84 / 18,752 (< 1 %)
Total registers	84
Total pins	35 / 315 (11 %)
Total virtual pins	0
Total memory bits	168 / 239,616 (< 1 %)
Embedded Multiplier 9-bit elements	2 / 52 (4 %)
Total PLLs	0 / 4 (0 %)

Fig.12 exponential accelerator synthesis report

	Status	From	To	Assignment Name	Value	Enabled	Entity	Comment	Tag
1	Ok		clk	Location	PIN_L1	Yes			
2	Ok		done	Location	PIN_U21	Yes			
3	Ok		empty	Location	PIN_Y22	Yes			
4	Ok		full	Location	PIN_U22	Yes			
5	Ok		pulse	Location	PIN_R22	Yes			
6	Ok		q[20]	Location	PIN_R17	Yes			
7	Ok		q[19]	Location	PIN_R18	Yes			
8	Ok		q[18]	Location	PIN_U18	Yes			
9	Ok		q[17]	Location	PIN_Y18	Yes			
10	Ok		q[16]	Location	PIN_Y19	Yes			
11	Ok		q[15]	Location	PIN_T18	Yes			
12	Ok		q[14]	Location	PIN_Y19	Yes			
13	Ok		q[13]	Location	PIN_U19	Yes			
14	Ok		q[12]	Location	PIN_R19	Yes			
15	Ok		q[11]	Location	PIN_R20	Yes			
16	Ok		rst	Location	PIN_L22	Yes			
17	Ok		start	Location	PIN_L21	Yes			
18	Ok		ui[1]	Location	PIN_V12	Yes			
19	Ok		ui[0]	Location	PIN_M22	Yes			
20	Ok		vi[4]	Location	PIN_M1	Yes			
21	Ok		vi[3]	Location	PIN_M2	Yes			
22	Ok		vi[2]	Location	PIN_U11	Yes			
23	Ok		vi[1]	Location	PIN_U12	Yes			
24	Ok		vi[0]	Location	PIN_W12	Yes			
25		<<new>>	<<new>>	<<new>>					

Fig.13 pin assignment