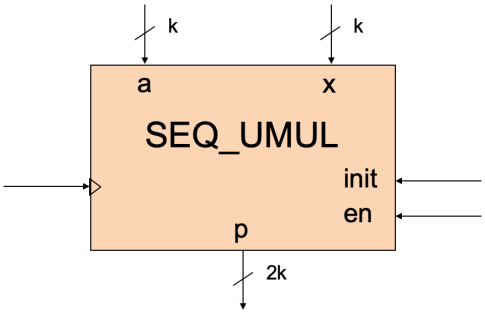
Lab 3
Implementing & Synthesizing Sequential Logic in VHDL

Main Task: Sequential Unsigned Multiplier (SEQ_UMUL)

A Sequential Unsigned Multiplier (SEQ_UMUL), implementing the Shift-and-Add algorithm, is defined below.

Interface:



Notation:
a: k-bit multiplicand
x: k-bit multiplier
p: 2k-bit product

In order to perform a multiplication, inputs a and x must be provided in the clock cycle 0, and then the following sequence of control signals must be applied:

Clock Cycles	init	en
0	1	1
1..k	0	1

Develop a Register-Transfer Level (RTL) VHDL description of SEQ_UMUL based on the block diagram shown in Fig. 1. In this diagram, the following notation is used:

- REG: Register,
- SRR: Shift-register with a parallel input, shifting right
- AND: k x 1 AND
- ADD: Unsigned Adder

Write a testbench capable of verifying the operation of the circuit for all possible values of the input pair (a, x), with k=4.
Synthesize, map, place, and route your circuit for k=4. Verify its operation using post-synthesis functional simulation, post-synthesis timing simulation, and post-implementation timing simulation.

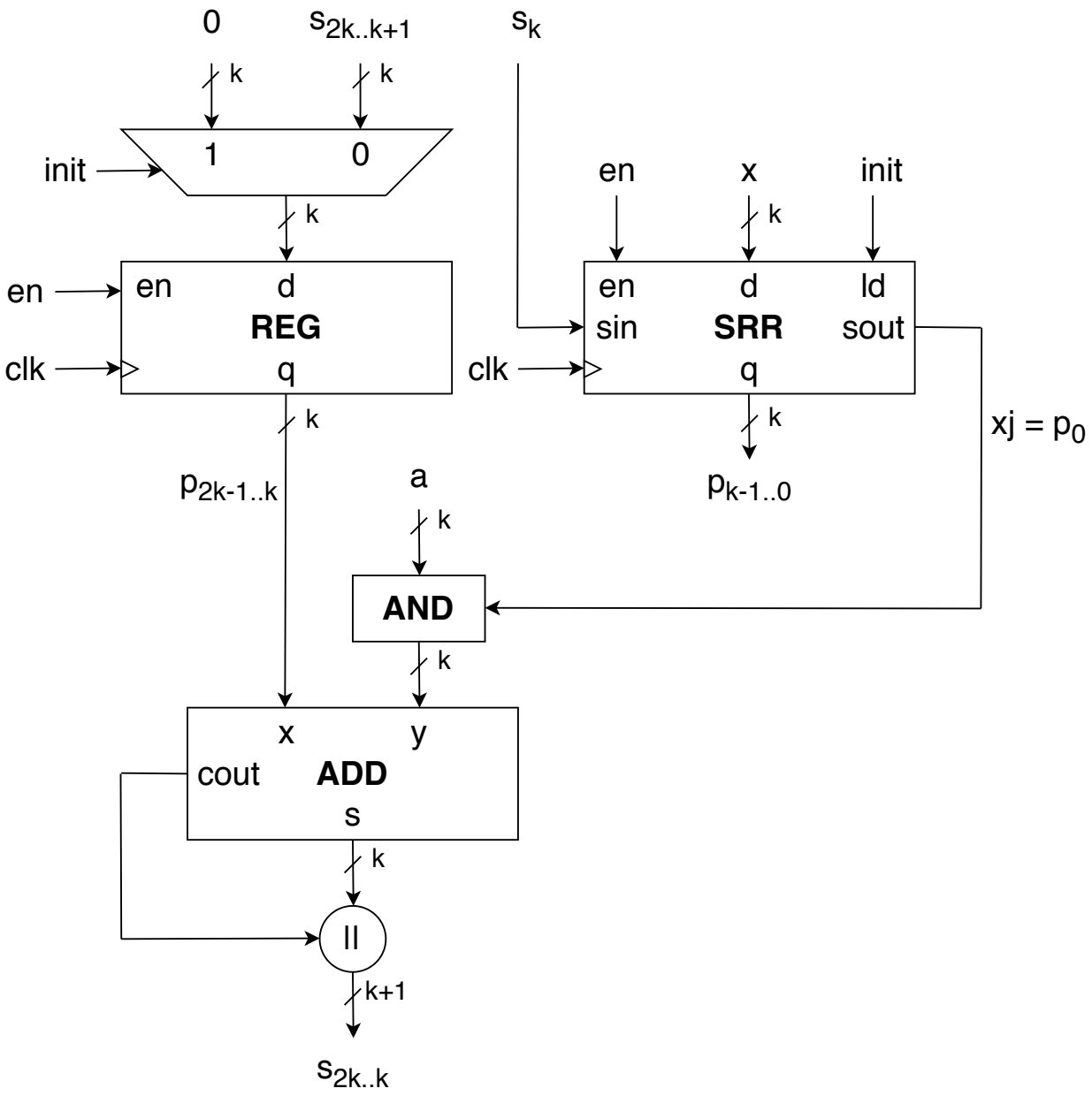


Fig. 1: Block diagram of a Sequential Unsigned Multiplier (SEQ_UMUL), implementing the Shift-and-Add algorithm.

Deliverables:

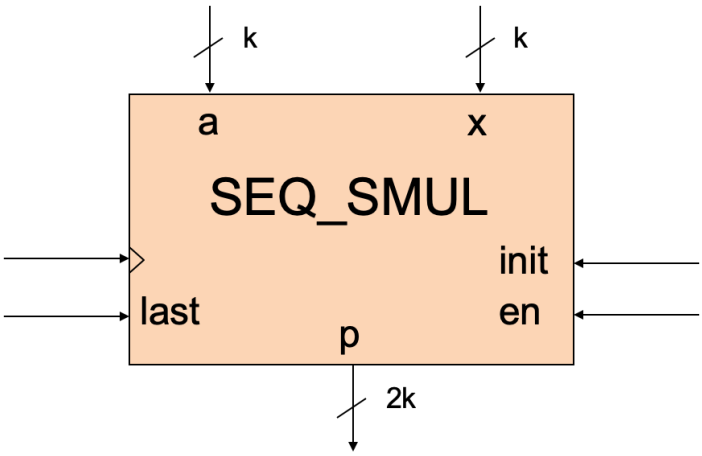
1. Synthesizable VHDL code of the SEQ_UMUL unit.
2. Corresponding testbench capable of verifying this code for correct operation using
    a. All combinations of inputs
    b. Expected output calculated in VHDL
    c. For incorrect results – inputs, expected output, and actual output displayed on the screen

3. Reports (saved as .txt) and schematic (saved as .pdf) obtained as a result of synthesis and implementation. Determine the number of slice LUTs and slice registers used by your circuit. Determine the Worst Negative Slack for the frequency of 100 MHz.
4. Post-synthesis model in VHDL generated using the Tcl command:
   write_vhdl -mode funcsim <output_folder>
5. Waveforms, in PDF, obtained by using post-synthesis functional simulation to verify results for at least 16 different values of inputs a and x, for k=4. These waveforms should show at least the following signals in your circuit: a, x, p, xj, $s_{2k..k}$.
6. Waveforms, in PDF, illustrating the difference among
   a. Post-synthesis functional simulation
   b. Post-synthesis timing simulation
   c. Post-implementation timing simulation.

   Explain the observed differences.

Bonus Tasks

1. Modify the block diagram shown in Fig. 1, so the circuit can support signed multiplication of k-bit integers expressed using the 2's complement representation.
   Hint: You can use the draw.io block diagram of SEQ_UMUL_k, available on the Lab Lecture 3 page.
   Assume the following modified interface:



In order to perform a multiplication, inputs a and x must be provided in the clock cycle 0, and then the following sequence of control signals must be applied:

Clock Cycles	init	en	last
0	1	1	0
1..k-1	0	1	0
k	0	1	1

2. Implement SEQ_SMUL in RTL VHDL.
3. Convert the testbench for SEQ_UMUL to the testbench for SEQ_SMUL, capable of verifying an output p for at least 5 different input pairs (a, x), selected as follows:
   $\{a \geq 0, x \geq 0\}$, $\{a < 0, x \geq 0\}$, $\{a \geq 0, x < 0\}$, $\{a < 0, x < 0\}$, $\{a \neq 0, x = 0\}$, with k=4.

4. Verify your implementation using the developed testbench.
5. Synthesize, map, place, and route your circuit for k=4. Verify its operation using post-synthesis functional simulation, post-synthesis timing simulation, and post-implementation timing simulation.

Deliverables:

1. Block diagram of your modified shift-and-add multiplier for signed integers
2. Synthesizable VHDL code of the SEQ_SMUL unit.
3. Corresponding testbench capable of verifying this code for correct operation using
    a. All combinations of inputs
    b. Expected output calculated in VHDL
    c. For incorrect results – inputs, expected output, and actual output displayed on the screen
4. Reports (saved as .txt) and schematic (saved as .pdf) obtained as a result of synthesis and implementation. Determine the number of slice LUTs and slice registers used by your circuit. Determine the Worst Negative Slack for the frequency of 100 MHz.
5. Post-synthesis model in VHDL generated using the Tcl command:
   write_vhdl -mode funcsim <output_folder>
6. Waveforms, in PDF, obtained by using post-synthesis functional simulation to verify results for at least 5 for at least 5 different input pairs (a, x), selected as follows:
   $\{a \geq 0, x \geq 0\}$, $\{a < 0, x \geq 0\}$, $\{a \geq 0, x < 0\}$, $\{a < 0, x < 0\}$, $\{a \neq 0, x = 0\}$, with k=4.
7. Waveforms, in PDF, illustrating the difference among
    a. Post-synthesis functional simulation
    b. Post-synthesis timing simulation
    c. Post-implementation timing simulation.

   Explain the observed differences.

Important Dates

	Friday Section	**Monday Section**
Hands-on Session and Introduction to the Experiment	Friday 02/10/2023, 8:40-11:20 AM	Monday 02/13/2023, 9:00-11:40 AM
Deliverables Due	**Monday** **02/20/2023, 11:59 PM**	**Monday** **02/20/2023, 11:59 PM**
Demo and Q&A	**Wed-Fri** **02/22-25/2023**	**Wed-Fri** **02/22-25/2023**