Design of Single-Precision Floating-Point Adders and Multipliers

Handout: 2024/09/24

Due: thr: 3 weeks later

1. FXP adder
   Design a 32-bit fixed-point (FXP) signed adder named as FXP_adder that performs addition D=A+B where A, B are 32-bit signed FXP inputs and D is the 32-bit FXP output. Use Synopsys Design Compiler (DC) to synthesize the Register-Transfer-Level (RTL) design into gate-level netlists based on a specific process technology. What is the critical path delay? What is the maximum working frequency? What is the total area? What is the total power and the corresponding frequency? Since the DC logic synthesis depends on user-specified constraints, you should synthesize your design with at least three different constraints: *delay-optimization*, *area-optimization*, and *in-between* (with reasonable delay constraint). Compare the differences of area/delay/power for the these three synthesis results. In general, the curve of area versus delay looks like a reciprocal curve. Note that total power consists of dynamic power and leakage power. Dynamic power is in general linearly proportional to the frequency while leakage power is usually independent of working frequency.

2. Non-pipelined and Pipelined FLP adders
   Design a IEEE 754 single-precision floating-point (FLP) adder that computes D=A+B where the two inputs A, B, and the output D are all in IEEE 754 single precision format.
   2.1 Design a non-pipelined fully combinational logic adder that consists of the following operations:
   - input data unpacking, including conversion from sign-magnitude to 2's complement
   - alignment
   - fixed-point (FXP) signed addition with enough guard bits
   - conversion to sign-magnitude format
   - normalization
   - rounding (including possible re-normalization of 1-bit right shift)
   - output result packing

   That is, all the above operations are executed in a single cycle. Name the design as FLP_adder. Write a testbench to verify the function of your register transfer level (RTL) design. Then use Synopsys Design Compiler (DC) to synthesize the RTL design to gate-level netlists based on a specific process technology. Verify the gate-level design again. What is the critical path delay? What is the total area? What is the toe total power? Note that after unpacking to extract mantissa, the numbers could be positive or negative with value between -2 and 2. Besides, after addition, the range of the sum is between -4 and 4. Thus, you should perform the alignment shift and the subsequent addition with operands represented in 2's complement format. Note that the signed results need at least 3 integer bits. The operand with smaller magnitude could be ignored if the exponent difference is 25 or more. Why?
   2.2 Pipeline the previous single-cycle design so that each of the above operations are executed in one cycle. Name this pipelined design as FLP_adder_pipe_7. That is the latency of the FLP_adder_pipeline_7 is seven cycles, but the throughput is one FLP addition per cycle. What is the critical path delay of each pipelined stage? What is the frequency of this pipelined design? What is the total area? What is the total power?
   2.3 Design another pipelined adder with four pipelined stages as following
   - Unpacking, alignment
   - FXP addition, conversion to sign-magnitude
   - Normalization
   - Rounding, packing

Name this pipelined design as FLP_adder_pipe_4. That is the latency of the FLP_adder_pipe_4 is four cycles, but the throughput is still one FLP addition per cycle. What is the critical path delay of each pipelined stage? What is the frequency of this pipelined design? What is the total area? What is the total power?

3. FXP multiplier
   Design a 32-bit fixed-point (FXP) signed multiplier named as FXP_multiplier that performs multiplication D=A*B where A, B are 32-bit FXP inputs and D is the 32 most significant bits (MSB) of the product P=A*B. What is the critical path delay? What is the working maximum frequency? What is the total area? What is the total power?

4. Non-pipelined and Pipelined multiplier
   Design a IEEE 754 single-precision floating-point (FLP) multiplier that computes D=A*B where the two inputs A, B, and the output D are all in IEEE 754 single precision format.
   4.1 Design a non-pipelined fully combinational logic multiplier that consists of the following operations:
   - input data unpacking
   - fixed-point (FXP) signed multiplication
   - conversion to sign-magnitude format
   - normalization
   - rounding
   - output result packing

   That is, all the above operations are executed in one cycle. Name the design as FLP_multiplier. What is the critical path delay? What is the total area? What is the toe total power? Note that after unpacking and conversion to signed representation, the inputs are between -2 and +2. And after FXP signed multiplication, the range of the product is between -4 and +4. Thus, 3 integer bits are needed in the FXP format.
   4.2 Design a pipelined multiplier with four pipelined stages as following
   - Unpacking (no alignment needed)
   - FXPmultiplication
   - Conversion to sign-magnitude
   - Normalization, rounding/re-normalization, packing (much simpler normalization)

   Name this pipelined design as FLP_multiplier_pipe_4. That is the latency of the FLP_multiplier_pipe_4 is four cycles, but the throughput is still one FLP multiplication per cycle. What is the critical path delay of each pipelined stage? What is the frequency of this pipelined design? What is the total area? What is the total power?

5. Comparison
   Use a table to compare all the above adder and multiplier designs with properly specified synthesis constraints. Provide your own comments on these designs. The comparison table looks like the following.
   In the table, the total area is divided into two parts: combinational logic and sequential logic (latches, flip-flops, and registers). The delay means critical path delay which is the reciprocal of maximum working frequency. The latency is the delay from the primary inputs to the primary outputs, i.e., the sum of the pipelined delays in all the pipeline stages in pipelined designs. Note that total power consists of dynamic power and leakage power. Dynamic power is in general linearly proportional to the frequency while leakage power is usually independent of working frequency.

		Area (um^2)			Delay	Latency	Power (W)		
		CL	SL	Total	(ns)	(ns)	dynamic	leakage	total
FXP_adder	area								
	mid								
	delay								
FLP_adder	area								
	mid								
	delay								
FLP_adder_7	area								
	mid								
	delay								
FLP_adder_4	area								
	mid								
	delay								
FXP_multiplier	area								
	mid								
	delay								
FLP_multiplier	area								
	mid								
	delay								
FLP_multiplier_4	area								
	mid								
	delay								

6. Upload your report to NSYSU course website with proper file names showing your student ID and homework number. Example: M1130400xx_ALU_HW1 . The attached compressed file should include all HDL codes, results of simulation, results of synthesis, final report, and all the supporting documents

7. 繳交檔案 & 配分：
   - MXXXXXXXXX_ALU_HW1
     - RTL (40%)
       - FLP_adder (8%)
         Top module and all submodule (.v)
       - FLP_mul (8%)
       - FLP_adder_7 (8%)
       - FLP_adder_4 (8%)
       - FLP_mul_4 (8%)
     - SYN (30%)
       - FLP_adder (6%)

- area (2%)
  all output file form design compiler (.v, .sdc, .sdf, .ddc, area, timing, power)
- delay (2%)
- mid (2%)

- FLP_mul (6%)
- FLP_adder_7 (6%)
- FLP_adder_4 (6%)
- FLP_mul_4 (6%)

- Report.ppt / .pdf (30%)
  - 架構圖 & pipeline (10%)
  - 各階 pipeline 數據 (10%)
  - Area/Delay/Powe 數據比較表格 (10%)

將請依照上述資料夾結構整理檔案，壓縮成 MXXXXXXXXX.zip 並繳交