NTHU- EE525500 Design of Chip Security- Spring 2024

Lab01 Sequential Circuit

Objective

Learn to use sequential logic to implement a message schedule circuit of SHA2-256

Prior Knowledge

1. Syntax of combinational and sequential circuits in Verilog.
2. Secure Hash Algorithm Standard (SHA) (FIPS 180-2).

Submission Content

1. Design a module that can:
    a. Calculate the correct outputs and comply with the IO protocol
    b. Pass the lint check (No warnings and errors)
    c. Be synthesized (No warnings and errors)
    d. NOT contain latches or combination loop after synthesis
2. A Report:
    a. Explain the design architecture and throughput.
    b. Explain the testbench (how to verify the design module)
    c. If necessary, use Wavedrom to draw the waveform

Module Specification

You need to implement an algorithm "**Message Schedule**" in SHA2

SHA-256 may be used to hash a message, M, having a length of ℓ bits, where where $0 \le \ell < 2^{64}$. The algorithm uses (1) a message schedule of sixty-four 32-bit words, (2) eight working variables of 32-bit each, and (3) a hash value of eight 32-bit words. The final result of SHA-256 is a 256-bit message digest.

After preprocessing is completed, each message block, $M^{(1)}$, $M^{(2)}$, …, $M^{(N)}$, is processed in order, **using the following steps to implement message schedule**:

$$W_t = \begin{cases} M_t^{(i)} & 0 \le t \le 15 \\ \sigma_1^{\{256\}}(W_{t-2}) + W_{t-7} + \sigma_0^{\{256\}}(W_{t-15}) + W_{t-16} & 16 \le t \le 63 \end{cases}$$

For more detailed algorithm, please refer to *https://csrc.nist.gov/files/pubs/fips/180-2/final/docs/fips180-2.pdf*

A message schedule module specification

Module name: sha2_msg

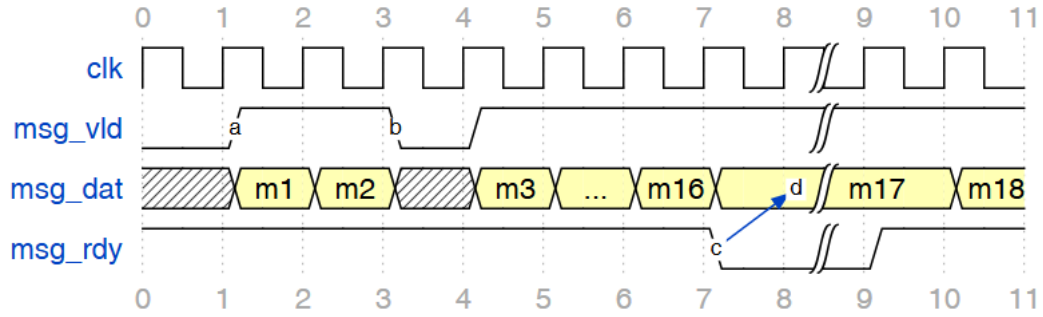
IO signals:

input	msg_vld	
input [31:0]	msg_dat	
output	msg_rdy	
output	word_vld	
output [31:0]	word	
input	word_rdy	
input	clk	//Periodic clock, 50MHz
input	rst_n	//Asynchronous reset, active low

The IO protocol of message-side is shown below:

1. rst_n will be active at the beginning of the test
2. Total message size is N * 512 bits
3. (a) msg_vld may rise at any time, indicating that msg_dat is valid
4. No matter what msg_rdy is, (b) msg_vld is allowed not to be continuous
5. If (c) msg_rdy is low, (d) msg_dat will remain unchanged until msg_rdy rises again
6. The msg_rdy should not wait for msg_vld is high before raising

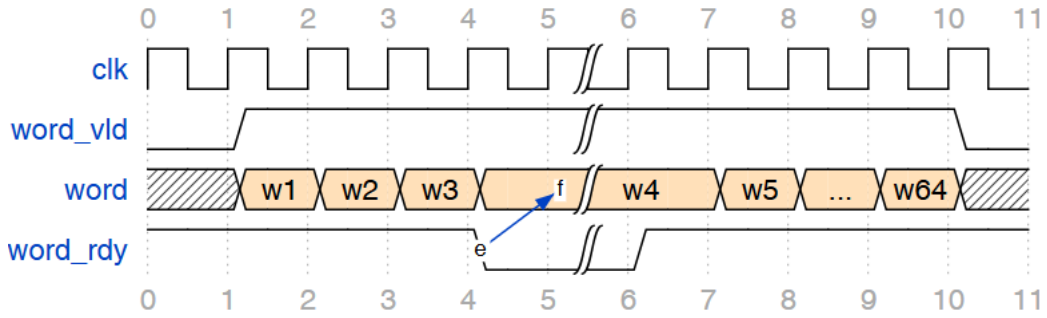
[Fig.1] Message Interface



The IO protocol of word-side is shown below:

7. If (e) word_rdy is low, (f) word must remain unchanged until word_rdy rises again

[Fig.2] Word Interface



Message Schedule Example

The following is one of the sets of messages and the corresponding word

- Message (512bits)

61626380	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000018

- Word (2048bits)

61626380	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000018
61626380	000f0000	7da86405	600003c6
3e9d7b78	0183fc00	12dcbfdb	e2e2c38e
c8215c1a	b73679a2	e5bc3909	32663c5b
9d209d67	ec8726cb	702138a4	d3b7973b
93f5997f	3b68ba73	aff4ffc1	f10a5c62
0a8b3996	72af830a	9409e33e	24641522
9f47bf94	f0a64f5a	3e246a79	27333ba3
0c4763f2	840abf27	7a290d5d	065c43da
fb3e89cb	cc7617db	b9e66c34	a9993667
84badedd	c21462bc	1487472c	b20f7a99
ef57b9cd	ebe6b238	9fe3095e	78bc8d4b
a43fcf15	668b2ff8	eeaba2cc	12b1edeb

Reference Environment

- Lab01.zip
  It contains a folder, sha2_msg/, which is the development environment for the module.
- If necessary, you can add your own submodules and test files into the environment. You can update ./sim/file.v or ./rtl/dut_include.v
- Please DO NOT:
  - Delete any files that were originally in the Lab01.zip
  - Change the file hierarchy (do not change the location of the files)
  - Increase or decrease the module's IO or change its width

Submit

- Lab01_xxxxxxx.zip (xxxxxxx is your student ID)
  Please run ./09_clean before you deliver. Use the following command to package the lab01 folder.
  zip -r lab01_xxxxxxx.zip lab01
- Lab01_xxxxxxx.pdf
  Please write your report and submit it in pdf format

Credit

- Functional pass + Lint pass + Synthesis Pass (No latches or combinational loops) (70%)
- Design area, small area is preferred (20%)
- Report (10%)