ECE 564 – PROJECT

SHA 256 – A CRYPTOGRAPHIC HASH FUNCTION

REPORT

by,

SHREYAS SRINIVASAN

STUDENT ID: 200255451

UNITY ID: ssrini22

Name: **Shreyas Srinivasan**

Unity-id: **ssrini22**

Student-ID: **200255421**

Delay (ns to run provided provided example). = **7627ns**

Clock period: **21.5ns**

# cycles”: 88+111+139 = **338**

(For message with length 1, 27, 55)

Logic Area: (um^2)

29036.56044

1/(delay.area) (ns^{-1}.um^{-2})

4.74 x 10^{-9}

Delay (TA provided example. TA to complete)

1/(delay.area) (TA)

Abstract:

The objective of this project is to design and simulate SHA 256 – A cryptographic Hash function using Verilog. A message of length varying from 1-55 characters is provided as the input from an SRAM and a corresponding Hashed output is obtained after required operations. The SHA-256 module is primarily designed using registers, adders and logic gates. An input message is operated for multiple iterations using the module components and a final output of 8x32 bits length, a cryptic version of the message is obtained which is written back to the SRAM.

Introduction:

The message will be contained as ascii in an SRAM and the length of the message is specified using a 6-bit number representing the number of ascii characters in the message. For example, if the message is 'hello' the length will be 6'd5 and the message SRAM will contain 0x68, 0x65, 0x6C, 0x6C, 0x6F

Steps followed in SHA-256 operation:

- The message is read and a 512-bit block/vector, M is constructed using the message and the bit length of the message. The vector M is then separated into an array of sixteen 32-bit words, M_1(0) .. M_1(15)
- The array M_1 is copied into the first 16 elements of a 64 32-bit word array, W. The elements 16 through 63 of W are processed using a combination of XOR and shift/rotate. Each element W[i] is a function of lower order elements e.g. W[i] = fn(W[i-2],W[i-7],W[i-15],W[i-16])

$$W_i = \sigma_1(Wi-2) + W_{i-7} + \sigma_0(W_{i-15}) + W_{i-16}, \quad 17 \leq i \leq 64$$

$$\sigma_0(X) = RotR(X, 7) \oplus RotR(X, 18) \oplus ShR(X, 3)$$

$$\sigma_1(X) = RotR(X, 17) \oplus RotR(X, 19) \oplus ShR(X, 10)$$

- We then construct another 64-element array, K by reading 64 values from the K SRAM.
- Once we have W and K, we load an eight 32-bit element array H from the contents of the H SRAM. We then initialize eight 32-bit registers, a, b, c, d, e, f, g, h from the contents of the H vector. We then iterate 64 times, j=0…63 on the a-h registers using element j from W and K and calculating a-h as,

$T1 = h + \Sigma1(e) + Ch(e, f, g) + Ki + Wi$	$T2 = \Sigma0(a) + Maj(a, b, c)$
$h = g$	$g = f$
$f = e$	$e = d + T1$
$d = c$	$c = b$
$b = a$	$a = T1 + T2$

$$Ch(X, Y, Z) = (X \wedge Y) \oplus (X \wedge Z)$$

$$Maj(X, Y, Z) = (X \wedge Y) \oplus (X \wedge Z) \oplus (Y \wedge Z)$$

$$\Sigma_0(X) = RotR(X, 2) \oplus RotR(X, 13) \oplus RotR(X, 22)$$

$$\Sigma_1(X) = RotR(X, 6) \oplus RotR(X, 11) \oplus RotR(X, 25)$$

We continue the above step for 64 iterations and on completion the final hash contents are calculated as,

$$H(t)\ 1 = H(t-1)\ 1 + a$$

$$H(t)\ 2 = H(t-1)\ 2 + b$$

$$H(t)\ 3 = H(t-1)\ 3 + c$$

$$H(t)\ 4 = H(t-1)\ 4 + d$$

$$H(t)\ 5 = H(t-1)\ 5 + e$$

$$H(t)\ 6 = H(t-1)\ 6 + f$$

$$H(t)\ 7 = H(t-1)\ 7 + g$$

$$H(t)\ 8 = H(t-1)\ 8 + h$$

$$\boxed{H = H(N)\ 1 \parallel H(N)\ 2 \parallel H(N)\ 3 \parallel H(N)\ 4 \parallel H(N)\ 5 \parallel H(N)\ 6 \parallel H(N)\ 7 \parallel H(N)}$$

The output consists of eight 32-bit vectors and the length is fixed for any input message length. The module is verified for different messages and the obtained output from the Verilog module is checked against a standard output obtained from a reference function.

Micro-architecture:

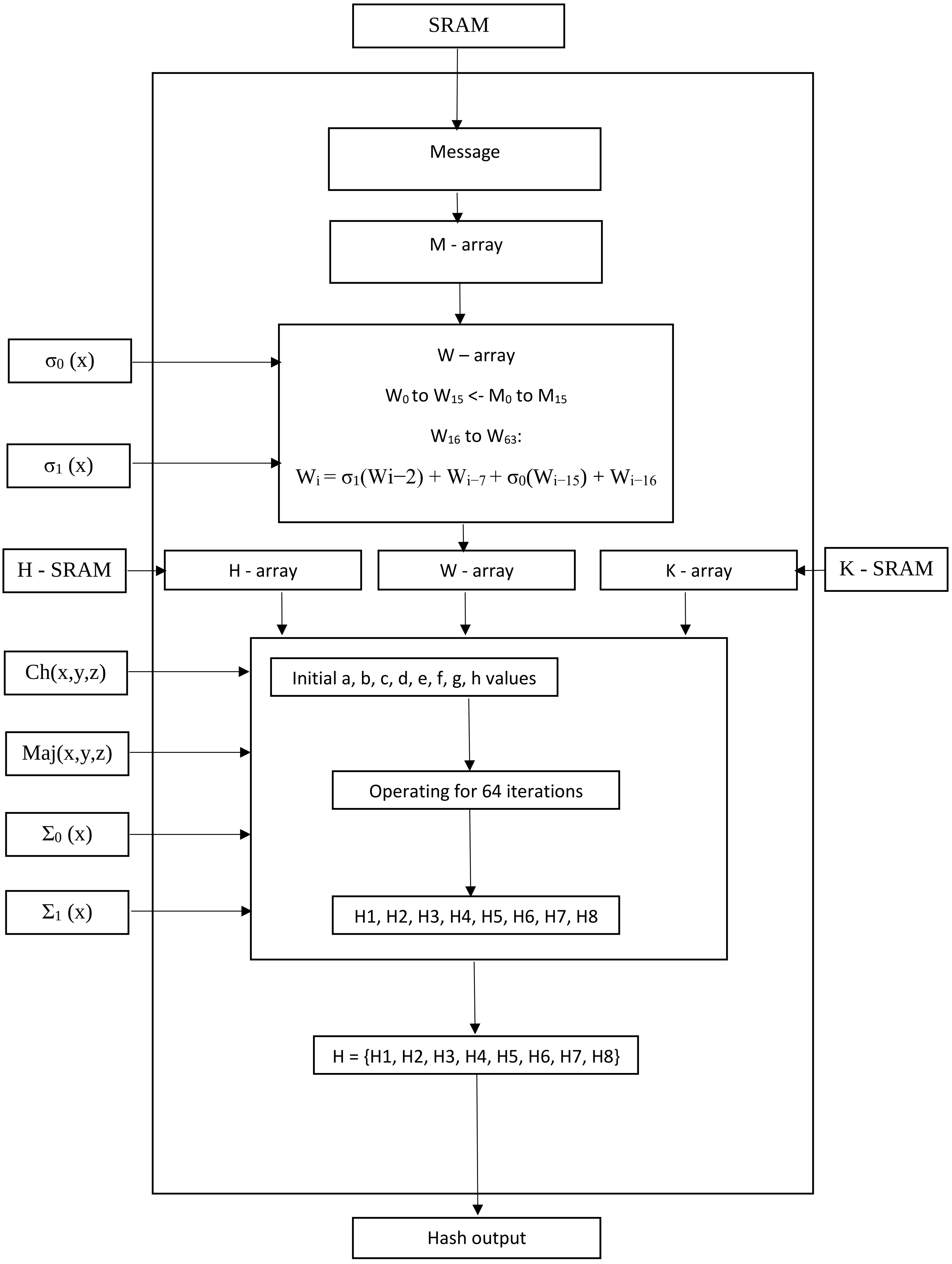
Inputs from SRAM:

- Input message
- Message length
- Enable & Write control signals
- Go signal
- K-vector
- H-vector

Output to SRAM:

- Hashed output message

Block Diagram and data flow:


SRAM
Message
M - array
W – array
W_0 to W_{15} <- M_0 to M_{15}
W_{16} to W_{63}:
$W_i = \sigma_1(W_{i-2}) + W_{i-7} + \sigma_0(W_{i-15}) + W_{i-16}$
σ_0 (x)
σ_1 (x)
H - SRAM
H - array
W - array
K - array
K - SRAM
Ch(x,y,z)
Maj(x,y,z)
Σ_0 (x)
Σ_1 (x)
Initial a, b, c, d, e, f, g, h values
Operating for 64 iterations
H1, H2, H3, H4, H5, H6, H7, H8
H = {H1, H2, H3, H4, H5, H6, H7, H8}
Hash output

Modules used:

Top module – For testbench & instantiating other modules

My design module – Work module and contains SHA-256 operation

SRAM module – For reading in and writing out values

Signals used:

Signal	Width(bits)	Description
Control		
xxx__dut__go	1	Starting the operation
xxx_dut__msg_length	log2(msg_length)	Length of message in bytes
dut__xxx__finish	1	Finishing the operation
Message Memory		
dut__msg__address	log2(msg_length)	Address of input message
dut__msg__enable	1	Enable signal for input message
dut__msg__write	1	Write signal for input message
dut__msg__data	8	Message in bits
K Memory		
dut__kmem__address	log2(No. of Ks)	Address of K input
dut__ kmem __enable	1	Enable signal for K input
dut__ kmem __write	1	Write signal for K input
dut__ kmem __data	32	K vector in bits
H Memory		
dut__hmem__address	log2(No. of Hs)	Address of H input
dut__ hmem __enable	1	Enable signal for H input
dut__ hmem __write	1	Write signal for H input
dut__ hmem __data	32	H vector in bits
Output Memory		
dut__dom__address	log2(Output length)	Address of output
dut__ dom __enable	1	Enable signal output
dut__ dom __write	1	Write signal for output
dut__ dom __data	32	Output vector in bits
Registers Used		
msg_in	8	Registering the input
msg	512	For loading initial message from SRAM
w	64x32	W-memory
k	32	K-vector value loaded and used for each iteration

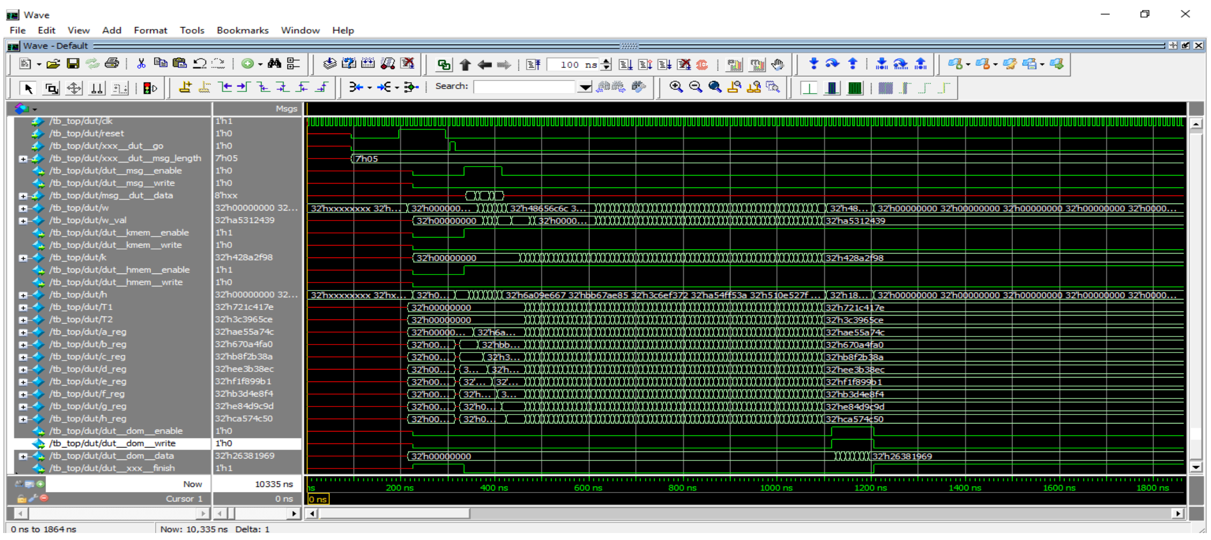
h	32	H – vector value loaded for 1st iteration and used for storing the final hash value
a_reg	32	Registers used for multiple-iteration operation
b_reg	32	
c_reg	32	
d_reg	32	
e_reg	32	
f_reg	32	
g_reg	32	
h_reg	32	
T1	32	
T2	32	
address	log2(msg_length)	Register used for accessing and incrementing the address from SRAM for loading input message
k_address	log2(No. of Ks)	Register used for accessing and incrementing the address from SRAM for loading K values
h_address	log2(No. of Hs)	Register used for accessing and incrementing the address from SRAM for loading H values
output_address	3	Register used for accessing and incrementing the address from SRAM for loading output message
length_count	log2(msg_length)	Register used for comparing length of message while loading
k_length	log2(No. of Ks)	Register used for comparing length of K value while loading
h_length	log2(No. of Hs)	Register used for comparing length of H value while loading
output_length_count	5	Register used for comparing length of output while loading
i	8	Used for iterations
j	8	
m	8	
n	8	
Iteration	8	
start	1	Control signals to indicate the successive operation that
start_w	1	

start_k	1	message, w-vector, k-vector are ready for usage
done, done1	1	Control signal to indicate finish of computing hash values
control_for_go	1	Control signal used to neglect intermediate ‘go’ signals during computation
compute_k	1	Control signal to signal start of computation of k value
k_first	1	Control signals used to signal the design to load values from SRAM or load values to SRAM
h_first	1	
comp_first	1	
out_first	1	
rst	1	Registering input signals
go	1	
finish	1	
dut_msg_length	log2(msg_length)	
msg_enable	1	Registering output signals
msg_write	1	
k_enable	1	
k_write	1	
h_enable	1	
h_write	1	
out_enable	1	
out_write	1	

Output Waveform:

Message length: 5 characters

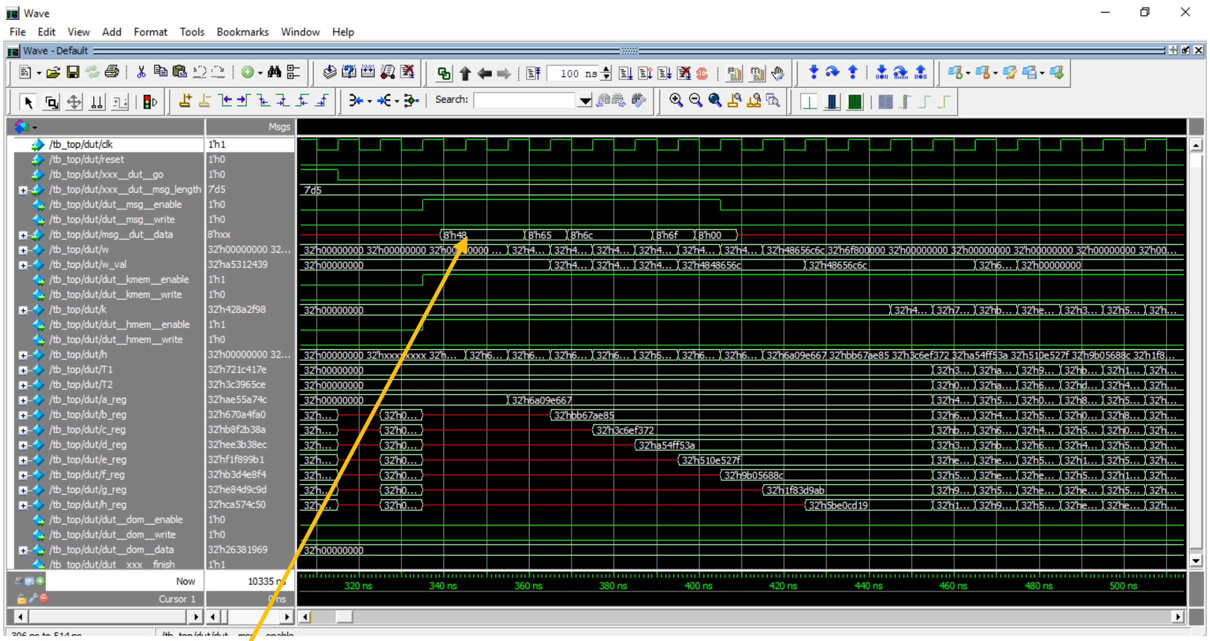
Message: “Hello” – 48, 65, 6c, 6c, 68



```
/tb_top/dut/clk                      1'h1
/tb_top/dut/reset                    1'h0
/tb_top/dut/xxx__dut__go             1'h0
/tb_top/dut/xxx__dut__msg_length     7'h05
/tb_top/dut/dut__msg__enable         1'h0
/tb_top/dut/dut__msg__write          1'h0
/tb_top/dut/msg__dut__data           8'hxx
/tb_top/dut/w                        32'h00000000 32...
/tb_top/dut/w_val                    32'ha5312439
/tb_top/dut/dut__kmem__enable        1'h1
/tb_top/dut/dut__kmem__write         1'h0
/tb_top/dut/k                        32'h428a2f98
/tb_top/dut/dut__hmem__enable        1'h1
/tb_top/dut/dut__hmem__write         1'h0
/tb_top/dut/h                        32'h00000000 32...
/tb_top/dut/T1                       32'h721c417e
/tb_top/dut/T2                       32'h3c3965ce
/tb_top/dut/a_reg                    32'hae55a74c
/tb_top/dut/b_reg                    32'h670a4fa0
/tb_top/dut/c_reg                    32'hb8f2b38a
/tb_top/dut/d_reg                    32'hee3b38ec
/tb_top/dut/e_reg                    32'hf1f899b1
/tb_top/dut/f_reg                    32'hb3d4e8f4
/tb_top/dut/g_reg                    32'he84d9c9d
/tb_top/dut/h_reg                    32'hca574c50
/tb_top/dut/dut__dom__enable         1'h0
/tb_top/dut/dut__dom__write          1'h0
/tb_top/dut/dut__dom__data           32'h26381969
/tb_top/dut/dut__xxx__finish         1'h1
                          Now        10335 ns
                     Cursor 1        0 ns
0 ns to 1864 ns      Now: 10,335 ns  Delta: 1
```

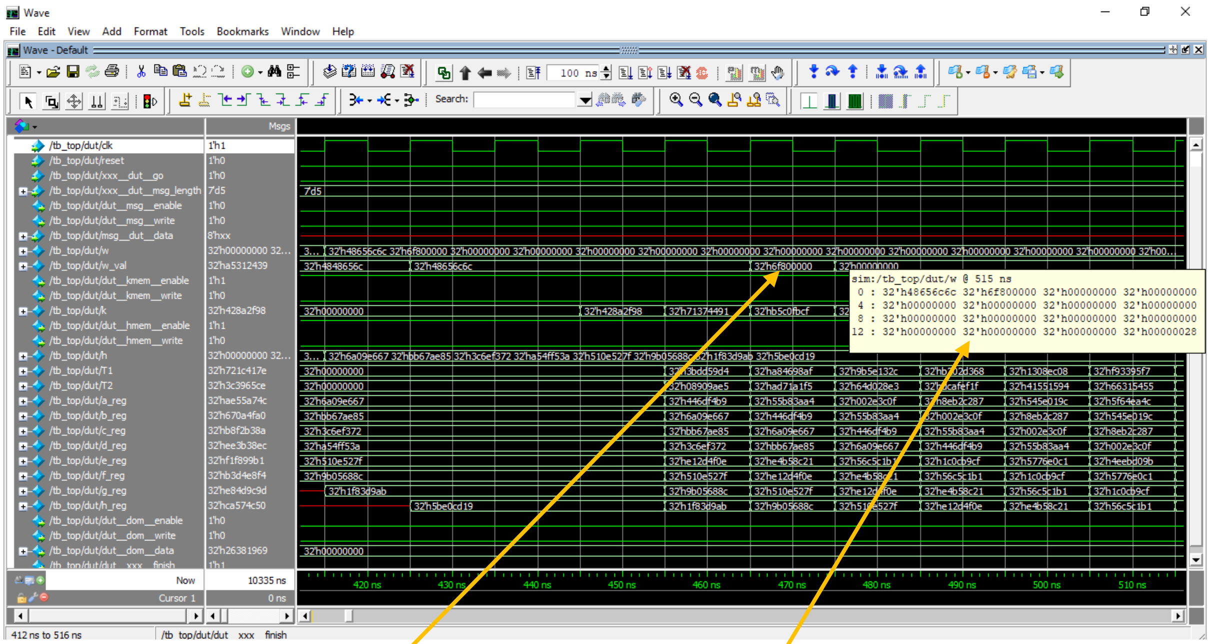
Complete Waveform

STEP 1:


Wave
File Edit View Add Format Tools Bookmarks Window Help
Wave - Default
/tb_top/dut/clk
/tb_top/dut/reset
/tb_top/dut/xxx__dut__go
/tb_top/dut/xxx__dut__msg_length
/tb_top/dut/dut__msg__enable
/tb_top/dut/dut__msg__write
/tb_top/dut/msg__dut__data
/tb_top/dut/w
/tb_top/dut/w_val
/tb_top/dut/dut__kmem__enable
/tb_top/dut/dut__kmem__write
/tb_top/dut/k
/tb_top/dut/dut__hmem__enable
/tb_top/dut/dut__hmem__write
/tb_top/dut/h
/tb_top/dut/T1
/tb_top/dut/T2
/tb_top/dut/a_reg
/tb_top/dut/b_reg
/tb_top/dut/c_reg
/tb_top/dut/d_reg
/tb_top/dut/e_reg
/tb_top/dut/f_reg
/tb_top/dut/g_reg
/tb_top/dut/h_reg
/tb_top/dut/dut__dom__enable
/tb_top/dut/dut__dom__write
/tb_top/dut/dut__dom__data
Now 10335 ns
Cursor 1

Input values getting loaded from SRAM after ‘go’ signal

STEP 2:

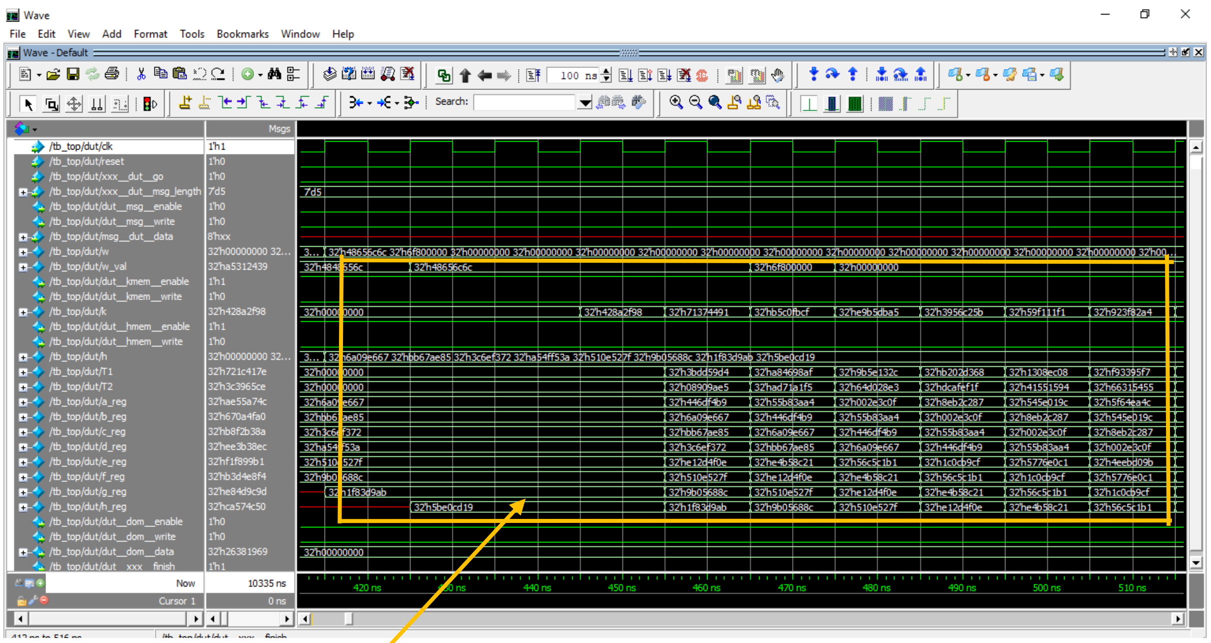


```
sim:/tb_top/dut/w @ 515 ns
 0 : 32'h48656c6c 32'h6f800000 32'h00000000 32'h00000000
 4 : 32'h00000000 32'h00000000 32'h00000000 32'h00000000
 8 : 32'h00000000 32'h00000000 32'h00000000 32'h00000000
12 : 32'h00000000 32'h00000000 32'h00000000 32'h00000028
```

W-value used for each iteration is extracted

32x16 W-vector computed

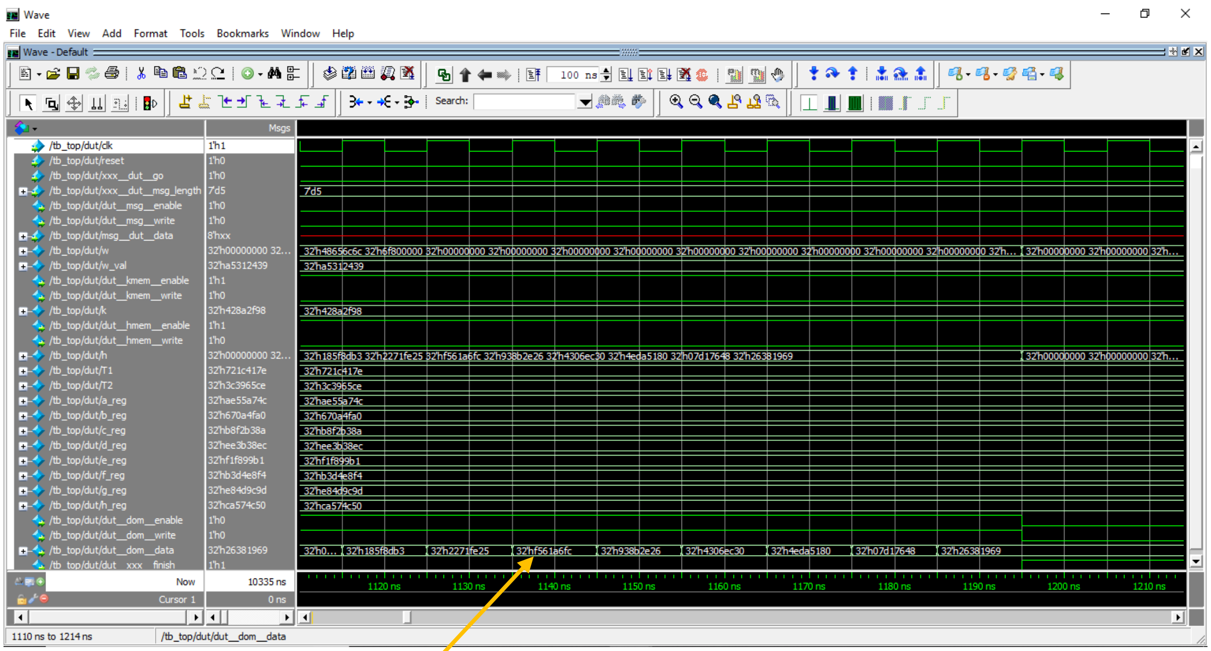
STEP 3:



```
/tb_top/dut/clk                       1'h1
/tb_top/dut/reset                     1'h0
/tb_top/dut/xxx__dut__go              1'h0
/tb_top/dut/xxx__dut__msg_length      7'd5
/tb_top/dut/dut__msg__enable          1'h0
/tb_top/dut/dut__msg__write           1'h0
/tb_top/dut/msg__dut__data            8'hxx
/tb_top/dut/w                         32'h00000000 32...
/tb_top/dut/w_val                     32'ha5312439
/tb_top/dut/dut__kmem__enable         1'h1
/tb_top/dut/dut__kmem__write          1'h0
/tb_top/dut/k                         32'h428a2f98
/tb_top/dut/dut__hmem__enable         1'h1
/tb_top/dut/dut__hmem__write          1'h0
/tb_top/dut/h                         32'h00000000 32...
/tb_top/dut/T1                        32'h721c417e
/tb_top/dut/T2                        32'h3c3965ce
/tb_top/dut/a_reg                     32'hae55a74c
/tb_top/dut/b_reg                     32'h670a4fa0
/tb_top/dut/c_reg                     32'hb8f2b38a
/tb_top/dut/d_reg                     32'hee3b38ec
/tb_top/dut/e_reg                     32'hf1f899b1
/tb_top/dut/f_reg                     32'hb3d4e8f4
/tb_top/dut/g_reg                     32'he84d9c9d
/tb_top/dut/h_reg                     32'hca574c50
/tb_top/dut/dut__dom__enable          1'h0
/tb_top/dut/dut__dom__write           1'h0
/tb_top/dut/dut__dom__data            32'h26381969
/tb_top/dut/dut__xxx__finish          1'h1
Now                                   10335 ns
Cursor 1                              0 ns
```

Iteration of 64 times, computing a, b, c, d, e, f, g, h on loading corresponding values from W-vector, K-vector & H-vector

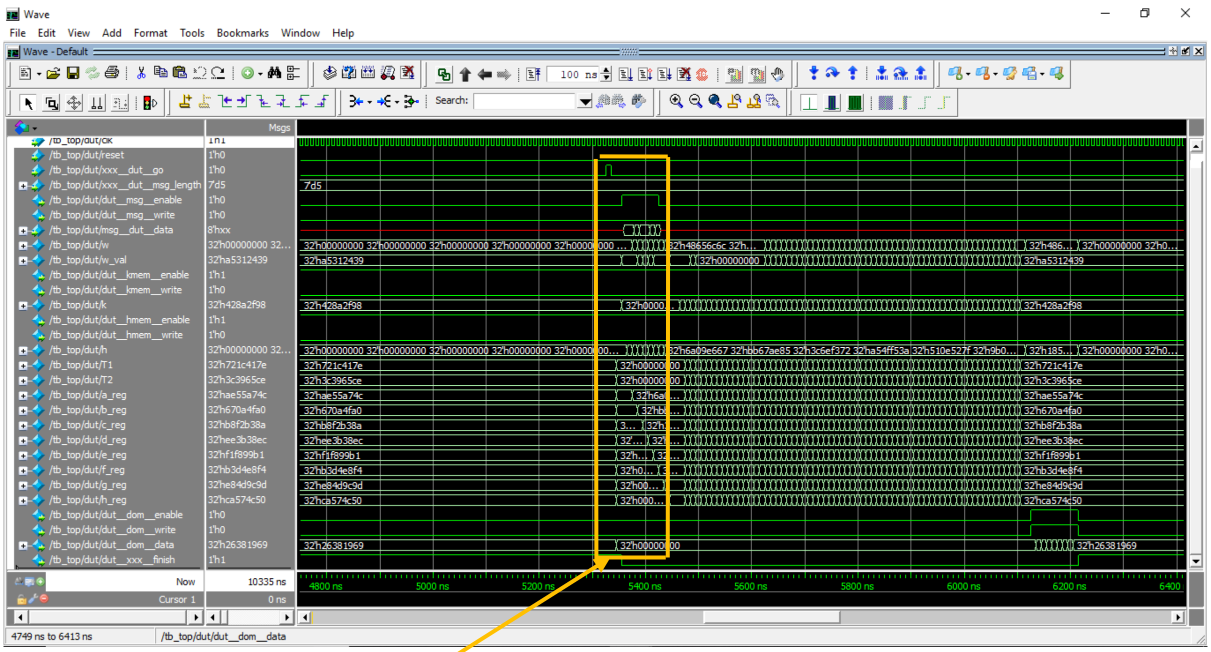
STEP 4:



```
/tb_top/dut/clk                       1'h1
/tb_top/dut/reset                     1'h0
/tb_top/dut/xxx__dut__go              1'h0
/tb_top/dut/xxx__dut__msg_length      7'd5
/tb_top/dut/dut__msg__enable          1'h0
/tb_top/dut/dut__msg__write           1'h0
/tb_top/dut/msg__dut__data            8'hxx
/tb_top/dut/w                         32'h00000000 32...
/tb_top/dut/w_val                     32'ha5312439
/tb_top/dut/dut__kmem__enable         1'h1
/tb_top/dut/dut__kmem__write          1'h0
/tb_top/dut/k                         32'h428a2f98
/tb_top/dut/dut__hmem__enable         1'h1
/tb_top/dut/dut__hmem__write          1'h0
/tb_top/dut/h                         32'h00000000 32...
/tb_top/dut/T1                        32'h721c417e
/tb_top/dut/T2                        32'h3c3965ce
/tb_top/dut/a_reg                     32'hae55a74c
/tb_top/dut/b_reg                     32'h670a4fa0
/tb_top/dut/c_reg                     32'hb8f2b38a
/tb_top/dut/d_reg                     32'hee3b38ec
/tb_top/dut/e_reg                     32'hf1f899b1
/tb_top/dut/f_reg                     32'hb3d4e8f4
/tb_top/dut/g_reg                     32'he84d9c9d
/tb_top/dut/h_reg                     32'hca574c50
/tb_top/dut/dut__dom__enable          1'h0
/tb_top/dut/dut__dom__write           1'h0
/tb_top/dut/dut__dom__data            32'h26381969
/tb_top/dut/dut__xxx__finish          1'h1
Now                                   10335 ns
Cursor 1                              0 ns
1110 ns to 1214 ns    /tb_top/dut/dut__dom__data
```

Output values loaded to SRAM and finish signal is asserted to make ready for next set of computation

STEP 5:


```
/tb_top/dut/clk                      1'h1
/tb_top/dut/reset                    1'h0
/tb_top/dut/xxx__dut__go             1'h0
/tb_top/dut/xxx__dut__msg_length     7'd5
/tb_top/dut/dut__msg__enable         1'h0
/tb_top/dut/dut__msg__write          1'h0
/tb_top/dut/msg__dut__data           8'hxx
/tb_top/dut/w                        32'h00000000 32...
/tb_top/dut/w_val                    32'ha5312439
/tb_top/dut/dut__kmem__enable        1'h1
/tb_top/dut/dut__kmem__write         1'h0
/tb_top/dut/k                        32'h428a2f98
/tb_top/dut/dut__hmem__enable        1'h1
/tb_top/dut/dut__hmem__write         1'h0
/tb_top/dut/h                        32'h00000000 32...
/tb_top/dut/T1                       32'h721c417e
/tb_top/dut/T2                       32'h3c3965ce
/tb_top/dut/a_reg                    32'hae55a74c
/tb_top/dut/b_reg                    32'h670a4fa0
/tb_top/dut/c_reg                    32'hb8f2b38a
/tb_top/dut/d_reg                    32'hee3b38ec
/tb_top/dut/e_reg                    32'hf1f899b1
/tb_top/dut/f_reg                    32'hb3d4e8f4
/tb_top/dut/g_reg                    32'he84d9c9d
/tb_top/dut/h_reg                    32'hca574c50
/tb_top/dut/dut__dom__enable         1'h0
/tb_top/dut/dut__dom__write          1'h0
/tb_top/dut/dut__dom__data           32'h26381969
/tb_top/dut/dut__xxx__finish         1'h1

Now        10335 ns
Cursor 1   0 ns

4749 ns to 6413 ns     /tb_top/dut/dut__dom__data
```

Vectors asserted to '0' on 'go' signal before next set of computation

Output Verification:

The output obtained from the Verilog module is checked for correctness on comparison with a reference Python function.

Scenarios Tested:

Message length = 1

Output from Python function	**Output from Verilog module**
H1: 0x4b68ab38	@00000000 4b68ab38
H2: 0x47feda7d	@00000001 47feda7d
H3: 0x6c62c1fb	@00000002 6c62c1fb
H4: 0xcbeebfa3	@00000003 cbeebfa3
H5: 0x5eab7351	@00000004 5eab7351
H6: 0xed5e78f4	@00000005 ed5e78f4
H7: 0xddadea5d	@00000006 ddadea5d
H8: 0xf64b8015	@00000007 f64b8015

Message length = 5

Output from Python function	**Output from Verilog module**
H1: 0x185f8db3	@00000000 185f8db3
H2: 0x2271fe25	@00000001 2271fe25
H3: 0xf561a6fc	@00000002 f561a6fc
H4: 0x938b2e26	@00000003 938b2e26
H5: 0x4306ec30	@00000004 4306ec30
H6: 0x4eda5180	@00000005 4eda5180
H7: 0x07d17648	@00000006 07d17648
H8: 0x26381969	@00000007 26381969

Message length = 10

Output from Python function	**Output from Verilog module**
H1: 0x36b45660	@00000000 36b45660
H2: 0xa055ed8c	@00000001 a055ed8c
H3: 0x725f4ad5	@00000002 725f4ad5
H4: 0x84f06eff	@00000003 84f06eff
H5: 0x9dbc75fe	@00000004 9dbc75fe
H6: 0x03a53704	@00000005 03a53704
H7: 0xbce79f80	@00000006 bce79f80
H8: 0x33d7e457	@00000007 33d7e457

Message length = 27

Output from Python function	**Output from Verilog module**
H1: 0xc0922a0b	@00000000 c0922a0b
H2: 0x27b7b58e	@00000001 27b7b58e
H3: 0x4ee4ddc7	@00000002 4ee4ddc7
H4: 0x4029f4c7	@00000003 4029f4c7
H5: 0xbc836ac3	@00000004 bc836ac3
H6: 0x29ce9f1c	@00000005 29ce9f1c

H7: 0x80cc48fe	@00000006 80cc48fe
H8: 0x996e9a26	@00000007 996e9a26

Message length = 55

Output from Python function	**Output from Verilog module**
H1: 0x7926e9b3	@00000000 7926e9b3
H2: 0xaffe2d46	@00000001 affe2d46
H3: 0x7beae296	@00000002 7beae296
H4: 0x6a0461d1	@00000003 6a0461d1
H5: 0x766c956c	@00000004 766c956c
H6: 0xcfc3bd39	@00000005 cfc3bd39
H7: 0xdeb71a5b	@00000006 deb71a5b
H8: 0xa8027e09	@00000007 a8027e09

Thus, the output from the Verilog module is verified and found to match that of the reference Python module.

Results achieved:

The design module is synthesized using synopsys and the timing and area reports are generated. The design is also optimized for area and clock period reduction.

Clock period set: 21.5ns

Report – Timing_max:

```
****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : MyDesign
Version: K-2015.06-SP1
Date   : Mon Nov 26 19:41:09 2018
****************************************

 # A fanout number of 1000 was used for high fanout net computations.

Operating Conditions: slow   Library:
NangateOpenCellLibrary_PDKv1_2_v2008_10_slow_nldm
Wire Load Model Mode: top
```

```
  Startpoint: iteration_reg[2]
              (rising edge-triggered flip-flop clocked by clk)
  Endpoint: e_reg_reg[31]
            (rising edge-triggered flip-flop clocked by clk)
  Path Group: clk
  Path Type: max

  Point                                                    Incr       Path
  -----------------------------------------------------------------------------
--
  clock clk (rise edge)                                  0.0000     0.0000
  clock network delay (ideal)                            0.0000     0.0000
  iteration_reg[2]/CK (DFF_X2)                           0.0000 #   0.0000
r
  iteration_reg[2]/Q (DFF_X2)                            0.6377     0.6377
f
  U23627/ZN (NOR2_X1)                                    0.2454     0.8831
r
  U19615/ZN (AND2_X4)                                    0.5477     1.4307
r
  U23280/ZN (AOI22_X1)                                   0.2376     1.6683
f
  U23626/ZN (AND4_X2)                                    0.4154     2.0837
f
  U23635/ZN (OR2_X4)                                     0.3273     2.4110
f
  U23632/ZN (NAND2_X2)                                   0.1185     2.5295
r
  add_1_root_add_0_root_add_1103_4_C903/A[0] (MyDesign_DW01_add_59)
                                                         0.0000     2.5295
r
  add_1_root_add_0_root_add_1103_4_C903/U1/ZN (AND2_X4)
                                                         0.1839     2.7134
r
  add_1_root_add_0_root_add_1103_4_C903/U1_1/S (FA_X1)
                                                         0.6923     3.4058
f
  add_1_root_add_0_root_add_1103_4_C903/SUM[1] (MyDesign_DW01_add_59)
                                                         0.0000     3.4058
f
  add_0_root_add_0_root_add_1103_4_C903/B[1] (MyDesign_DW01_add_57)
                                                         0.0000     3.4058
f
  add_0_root_add_0_root_add_1103_4_C903/U1_1/CO (FA_X1)
                                                         0.6478     4.0536
f
  add_0_root_add_0_root_add_1103_4_C903/U1_2/CO (FA_X1)
                                                         0.5172     4.5708
f
  add_0_root_add_0_root_add_1103_4_C903/U1_3/CO (FA_X1)
                                                         0.5172     5.0880
f
  add_0_root_add_0_root_add_1103_4_C903/U1_4/CO (FA_X1)
                                                         0.5172     5.6053
f
  add_0_root_add_0_root_add_1103_4_C903/U1_5/CO (FA_X1)
                                                         0.5172     6.1225
f
  add_0_root_add_0_root_add_1103_4_C903/U1_6/CO (FA_X1)
                                                         0.5172     6.6398
f
  add_0_root_add_0_root_add_1103_4_C903/U1_7/CO (FA_X1)
                                                         0.5172     7.1570
f
  add_0_root_add_0_root_add_1103_4_C903/U1_8/CO (FA_X1)
                                                         0.5172     7.6743
f
  add_0_root_add_0_root_add_1103_4_C903/U1_9/CO (FA_X1)
```

```
                                                      0.5172     8.1915
f
  add_0_root_add_0_root_add_1103_4_C903/U1_10/CO (FA_X1)
                                                      0.5172     8.7087
f
  add_0_root_add_0_root_add_1103_4_C903/U1_11/CO (FA_X1)
                                                      0.5172     9.2260
f
  add_0_root_add_0_root_add_1103_4_C903/U1_12/CO (FA_X1)
                                                      0.5172     9.7432
f
  add_0_root_add_0_root_add_1103_4_C903/U1_13/CO (FA_X1)
                                                      0.5172    10.2605
f
  add_0_root_add_0_root_add_1103_4_C903/U1_14/CO (FA_X1)
                                                      0.5172    10.7777
f
  add_0_root_add_0_root_add_1103_4_C903/U1_15/CO (FA_X1)
                                                      0.5172    11.2950
f
  add_0_root_add_0_root_add_1103_4_C903/U1_16/CO (FA_X1)
                                                      0.5172    11.8122
f
  add_0_root_add_0_root_add_1103_4_C903/U1_17/CO (FA_X1)
                                                      0.5172    12.3294
f
  add_0_root_add_0_root_add_1103_4_C903/U1_18/CO (FA_X1)
                                                      0.5172    12.8467
f
  add_0_root_add_0_root_add_1103_4_C903/U1_19/CO (FA_X1)
                                                      0.5172    13.3639
f
  add_0_root_add_0_root_add_1103_4_C903/U1_20/CO (FA_X1)
                                                      0.5172    13.8812
f
  add_0_root_add_0_root_add_1103_4_C903/U1_21/CO (FA_X1)
                                                      0.5172    14.3984
f
  add_0_root_add_0_root_add_1103_4_C903/U1_22/CO (FA_X1)
                                                      0.5172    14.9157
f
  add_0_root_add_0_root_add_1103_4_C903/U1_23/CO (FA_X1)
                                                      0.5172    15.4329
f
  add_0_root_add_0_root_add_1103_4_C903/U1_24/CO (FA_X1)
                                                      0.5172    15.9501
f
  add_0_root_add_0_root_add_1103_4_C903/U1_25/CO (FA_X1)
                                                      0.5172    16.4674
f
  add_0_root_add_0_root_add_1103_4_C903/U1_26/CO (FA_X1)
                                                      0.5172    16.9846
f
  add_0_root_add_0_root_add_1103_4_C903/U1_27/CO (FA_X1)
                                                      0.5172    17.5019
f
  add_0_root_add_0_root_add_1103_4_C903/U1_28/CO (FA_X1)
                                                      0.5172    18.0191
f
  add_0_root_add_0_root_add_1103_4_C903/U1_29/CO (FA_X1)
                                                      0.5172    18.5364
f
  add_0_root_add_0_root_add_1103_4_C903/U1_30/CO (FA_X1)
                                                      0.5172    19.0536
f
  add_0_root_add_0_root_add_1103_4_C903/U1_31/S (FA_X1)
                                                      0.7162    19.7698
r
  add_0_root_add_0_root_add_1103_4_C903/SUM[31] (MyDesign_DW01_add_57)
```

```
                                                          0.0000    19.7698
r
  U23817/Z (CLKBUF_X3)                                    0.1919    19.9616
r
  add_1103_C908/B[31] (MyDesign_DW01_add_0)               0.0000    19.9616
r
  add_1103_C908/U1_31/S (FA_X1)                           0.6980    20.6596
f
  add_1103_C908/SUM[31] (MyDesign_DW01_add_0)             0.0000    20.6596
f
  U23811/ZN (AOI22_X2)                                    0.2318    20.8914
r
  U23810/ZN (OAI21_X2)                                    0.1556    21.0470
f
  e_reg_reg[31]/D (DFF_X1)                                0.0000    21.0470
f
  data arrival time                                                 21.0470

  clock clk (rise edge)                                  21.5000    21.5000
  clock network delay (ideal)                             0.0000    21.5000
  clock uncertainty                                      -0.0500    21.4500
  e_reg_reg[31]/CK (DFF_X1)                               0.0000    21.4500
r
  library setup time                                     -0.3620    21.0880
  data required time                                                21.0880
  ----------------------------------------------------------------------
--
  data required time                                                21.0880
  data arrival time                                                -21.0470
  ----------------------------------------------------------------------
--
  slack (MET)                                                        0.0410
```

Report – Timing_max_slow_holdfixed:

```
****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : MyDesign
Version: K-2015.06-SP1
Date   : Mon Nov 26 19:41:28 2018
****************************************

 # A fanout number of 1000 was used for high fanout net computations.

Operating Conditions: slow   Library:
NangateOpenCellLibrary_PDKv1_2_v2008_10_slow_nldm
Wire Load Model Mode: top

  Startpoint: iteration_reg[2]
              (rising edge-triggered flip-flop clocked by clk)
  Endpoint: e_reg_reg[31]
            (rising edge-triggered flip-flop clocked by clk)
  Path Group: clk
  Path Type: max

  Point                                                     Incr       Path
  ----------------------------------------------------------------------
--
  clock clk (rise edge)                                   0.0000     0.0000
  clock network delay (ideal)                             0.0000     0.0000
  iteration_reg[2]/CK (DFF_X2)                            0.0000 #   0.0000
r
  iteration_reg[2]/Q (DFF_X2)                             0.6377     0.6377
f
```

```
  U23627/ZN (NOR2_X1)                                        0.2454     0.8831
r
  U19615/ZN (AND2_X4)                                        0.5477     1.4307
r
  U23280/ZN (AOI22_X1)                                       0.2376     1.6683
f
  U23626/ZN (AND4_X2)                                        0.4154     2.0837
f
  U23635/ZN (OR2_X4)                                         0.3273     2.4110
f
  U23632/ZN (NAND2_X2)                                       0.1185     2.5295
r
  add_1_root_add_0_root_add_1103_4_C903/A[0] (MyDesign_DW01_add_59)
                                                             0.0000     2.5295
r
  add_1_root_add_0_root_add_1103_4_C903/U1/ZN (AND2_X4)
                                                             0.1839     2.7134
r
  add_1_root_add_0_root_add_1103_4_C903/U1_1/S (FA_X1)
                                                             0.6923     3.4058
f
  add_1_root_add_0_root_add_1103_4_C903/SUM[1] (MyDesign_DW01_add_59)
                                                             0.0000     3.4058
f
  add_0_root_add_0_root_add_1103_4_C903/B[1] (MyDesign_DW01_add_57)
                                                             0.0000     3.4058
f
  add_0_root_add_0_root_add_1103_4_C903/U1_1/CO (FA_X1)
                                                             0.6478     4.0536
f
  add_0_root_add_0_root_add_1103_4_C903/U1_2/CO (FA_X1)
                                                             0.5172     4.5708
f
  add_0_root_add_0_root_add_1103_4_C903/U1_3/CO (FA_X1)
                                                             0.5172     5.0880
f
  add_0_root_add_0_root_add_1103_4_C903/U1_4/CO (FA_X1)
                                                             0.5172     5.6053
f
  add_0_root_add_0_root_add_1103_4_C903/U1_5/CO (FA_X1)
                                                             0.5172     6.1225
f
  add_0_root_add_0_root_add_1103_4_C903/U1_6/CO (FA_X1)
                                                             0.5172     6.6398
f
  add_0_root_add_0_root_add_1103_4_C903/U1_7/CO (FA_X1)
                                                             0.5172     7.1570
f
  add_0_root_add_0_root_add_1103_4_C903/U1_8/CO (FA_X1)
                                                             0.5172     7.6743
f
  add_0_root_add_0_root_add_1103_4_C903/U1_9/CO (FA_X1)
                                                             0.5172     8.1915
f
  add_0_root_add_0_root_add_1103_4_C903/U1_10/CO (FA_X1)
                                                             0.5172     8.7087
f
  add_0_root_add_0_root_add_1103_4_C903/U1_11/CO (FA_X1)
                                                             0.5172     9.2260
f
  add_0_root_add_0_root_add_1103_4_C903/U1_12/CO (FA_X1)
                                                             0.5172     9.7432
f
  add_0_root_add_0_root_add_1103_4_C903/U1_13/CO (FA_X1)
                                                             0.5172    10.2605
f
  add_0_root_add_0_root_add_1103_4_C903/U1_14/CO (FA_X1)
                                                             0.5172    10.7777
f
```

```
  add_0_root_add_0_root_add_1103_4_C903/U1_15/CO (FA_X1)
                                                          0.5172    11.2950
f
  add_0_root_add_0_root_add_1103_4_C903/U1_16/CO (FA_X1)
                                                          0.5172    11.8122
f
  add_0_root_add_0_root_add_1103_4_C903/U1_17/CO (FA_X1)
                                                          0.5172    12.3294
f
  add_0_root_add_0_root_add_1103_4_C903/U1_18/CO (FA_X1)
                                                          0.5172    12.8467
f
  add_0_root_add_0_root_add_1103_4_C903/U1_19/CO (FA_X1)
                                                          0.5172    13.3639
f
  add_0_root_add_0_root_add_1103_4_C903/U1_20/CO (FA_X1)
                                                          0.5172    13.8812
f
  add_0_root_add_0_root_add_1103_4_C903/U1_21/CO (FA_X1)
                                                          0.5172    14.3984
f
  add_0_root_add_0_root_add_1103_4_C903/U1_22/CO (FA_X1)
                                                          0.5172    14.9157
f
  add_0_root_add_0_root_add_1103_4_C903/U1_23/CO (FA_X1)
                                                          0.5172    15.4329
f
  add_0_root_add_0_root_add_1103_4_C903/U1_24/CO (FA_X1)
                                                          0.5172    15.9502
f
  add_0_root_add_0_root_add_1103_4_C903/U1_25/CO (FA_X1)
                                                          0.5172    16.4674
f
  add_0_root_add_0_root_add_1103_4_C903/U1_26/CO (FA_X1)
                                                          0.5172    16.9846
f
  add_0_root_add_0_root_add_1103_4_C903/U1_27/CO (FA_X1)
                                                          0.5172    17.5019
f
  add_0_root_add_0_root_add_1103_4_C903/U1_28/CO (FA_X1)
                                                          0.5172    18.0191
f
  add_0_root_add_0_root_add_1103_4_C903/U1_29/CO (FA_X1)
                                                          0.5172    18.5364
f
  add_0_root_add_0_root_add_1103_4_C903/U1_30/CO (FA_X1)
                                                          0.5172    19.0536
f
  add_0_root_add_0_root_add_1103_4_C903/U1_31/S (FA_X1)
                                                          0.7162    19.7698
r
  add_0_root_add_0_root_add_1103_4_C903/SUM[31] (MyDesign_DW01_add_57)
                                                          0.0000    19.7698
r
  U23817/Z (CLKBUF_X3)                                    0.1919    19.9616
r
  add_1103_C908/B[31] (MyDesign_DW01_add_0)               0.0000    19.9616
r
  add_1103_C908/U1_31/S (FA_X1)                           0.6980    20.6596
f
  add_1103_C908/SUM[31] (MyDesign_DW01_add_0)             0.0000    20.6596
f
  U23811/ZN (AOI22_X2)                                    0.2318    20.8914
r
  U23810/ZN (OAI21_X2)                                    0.1556    21.0470
f
  e_reg_reg[31]/D (DFF_X1)                                0.0000    21.0470
f
  data arrival time                                                 21.0470
  clock clk (rise edge)                              21.5000    21.5000
  clock network delay (ideal)                         0.0000    21.5000
  clock uncertainty                                  -0.0500    21.4500
  e_reg_reg[31]/CK (DFF_X1)                           0.0000    21.4500
r
  library setup time                                 -0.3367    21.1133
  data required time                                            21.1133
  -------------------------------------------------------------------------
--
  data required time                                            21.1133
  data arrival time                                            -21.0470
  -------------------------------------------------------------------------
--
  slack (MET)                                                    0.0663
```

Report – Timing_min_fast_holdchecked:

```
****************************************
Report : timing
        -path full
        -delay min
        -max_paths 1
Design : MyDesign
Version: K-2015.06-SP1
Date   : Mon Nov 26 19:41:21 2018
****************************************

 # A fanout number of 1000 was used for high fanout net computations.

Operating Conditions: fast   Library:
NangateOpenCellLibrary_PDKv1_2_v2008_10_fast_nldm
Wire Load Model Mode: top

  Startpoint: rst_reg (rising edge-triggered flip-flop clocked by clk)
  Endpoint: control_for_go_reg
            (rising edge-triggered flip-flop clocked by clk)
  Path Group: clk
  Path Type: min

  Point                                    Incr       Path
  -----------------------------------------------------------
  clock clk (rise edge)                  0.0000     0.0000
  clock network delay (ideal)            0.0000     0.0000
  rst_reg/CK (DFF_X2)                    0.0000 #   0.0000 r
  rst_reg/Q (DFF_X2)                     0.0560     0.0560 r
  U12878/ZN (NOR2_X2)                    0.0095     0.0656 f
  control_for_go_reg/D (DFF_X2)          0.0000     0.0656 f
  data arrival time                                 0.0656

  clock clk (rise edge)                  0.0000     0.0000
  clock network delay (ideal)            0.0000     0.0000
  clock uncertainty                      0.0500     0.0500
  control_for_go_reg/CK (DFF_X2)         0.0000     0.0500 r
  library hold time                      0.0006     0.0506
  data required time                                0.0506
  -----------------------------------------------------------
  data required time                                0.0506
  data arrival time                                -0.0656
  -----------------------------------------------------------
  slack (MET)                                       0.0150
```

Area Report:

```
****************************************
Report : area
Design : MyDesign
Version: K-2015.06-SP1
Date   : Mon Nov 26 19:51:48 2018
****************************************

Library(s) Used:

    NangateOpenCellLibrary_PDKv1_2_v2008_10_slow_nldm (File:
/afs/eos.ncsu.edu/lockers/research/ece/wdavis/tech/nangate/NangateOpenCellLibrary_PDKv1
_2_v2008_10/liberty/520/NangateOpenCellLibrary_PDKv1_2_v2008_10_slow_nldm.db)

Number of ports: 6393
Number of nets: 22989
Number of cells: 13376
Number of combinational cells: 11512
Number of sequential cells: 1796
Number of macros/black boxes: 0
Number of buf/inv: 1570
Number of references: 124

Combinational area: 20756.778260
Buf/Inv area: 862.106006
Noncombinational area: 8279.782185
Macro/Black Box area: 0.000000
Net Interconnect area: undefined (No wire load specified)

Total cell area: 29036.560444
```

Conclusion:

Thus, the SHA-256 hashing function is designed and simulated using Verilog. An 8x32 bit hash output is obtained for any message of length 1-55 characters. The obtained output is compared with a reference output from a python program for correctness and is verified using the same.

The design is synthesized using Synopsys and the timing and area reports are generated. The design is then optimized at coding level by removing unwanted and redundant components in-order to reduce the cell area of design and to achieve a lower clock period. The reports are then again generated, and the above step is continued to achieve the design constraint as low as possible.

Thus, the clock period and area achieved are reported and the various timing reports are also provided. It is found that the slack is met in every report and thus the setup and hold constraints are managed and achieved.

Clock period: 21.5ns

Area achieved: 29036.56044 µm^2

# cycles for message with 1 character: 88

# cycles for message with 27 character: 111

# cycles for message with 55 character: 139

Total cycles = 338

Overall performance metric = 21.5x 10^{-9} x 29036.56044 x 338

Overall performance metric = 211008685 ns µm^2