Resource-Efficient Implementation of BLUE MIDNIGHT WISH-256 Hash Function on Xilinx FPGA Platform

Mohamed El-Hadedy[1,2] , Martin Margala[2], Danilo Gligoroski[1], Svein Johan Knapskog[1]

Email hadedy@q2s.ntnu.no

[1]Centre for Quantifiable Quality of Service in Communication Systems (**Q2S**)

Norwegian University of Science and Technology (NTNU), Norway

[2]University of Massachusetts Lowell,MA,USA

Outline

- General characteristics of **B**lue **M**idnight **W**ish
- BMW-256 Hashing Core FPGA Architecture
- BMW-256 Hashing Core Operations
- FPGA Implementations Characteristics
- Conclusions

Outline

- **General characteristics of Blue Midnight Wish**
- BMW-256 Hashing Core FPGA Architecture
- BMW-256 Hashing Core Operations
- FPGA Implementations Characteristics
- Conclusions

General design characteristics for Blue Midnight Wish

Algorithm: BLUE MIDNIGHT WISH

Input: Message M of length l bits, and the message digest size n.

Output: A message digest *Hash*, that is n bits long.

1. Preprocessing
   (a) Pad the message M.
   (b) Parse the padded message into N, m-bit message blocks, $M^{(1)}, M^{(2)}, \ldots, M^{(N)}$.
   (c) Set the initial value of the double pipe $H^{(0)}$.
2. Hash computation

   For $i = 1$ to N

   {

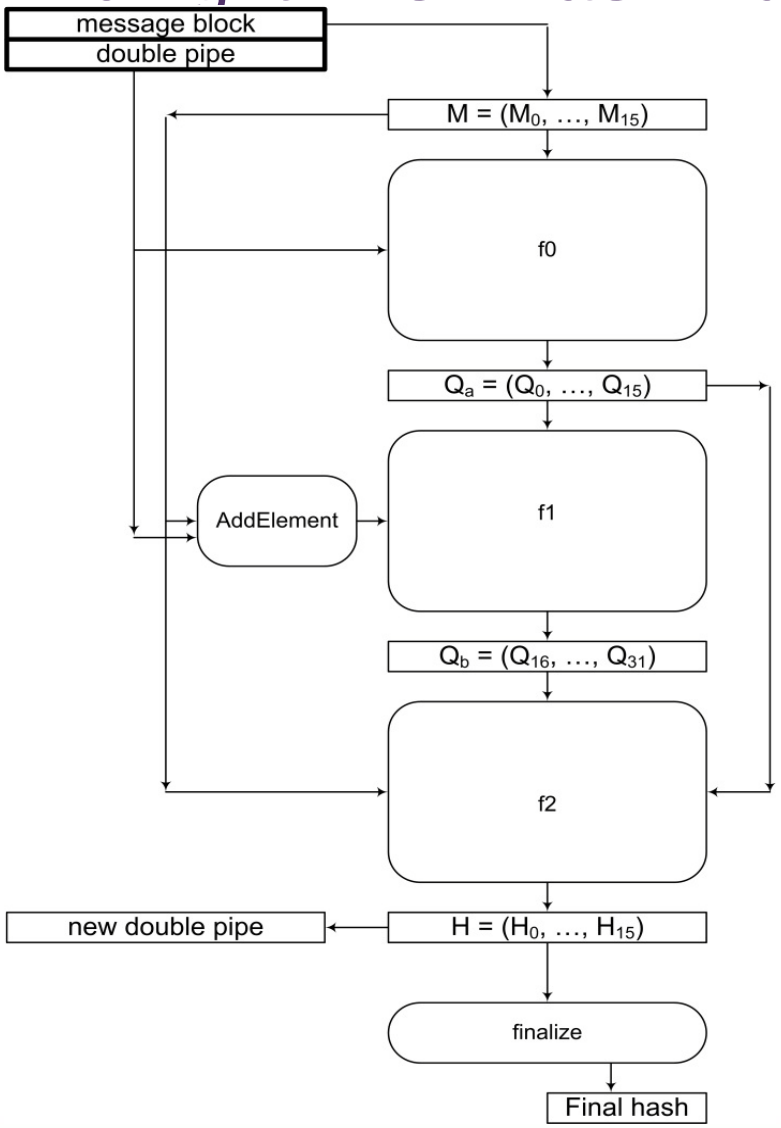
   $Q_a^{(i)} = f_0(M^{(i)}, H^{(i-1)});$

   $Q_b^{(i)} = f_1(M^{(i)}, Q_a^{(i)});$

   $H^{(i)} = f_2(M^{(i)}, Q_a^{(i)}, Q_b^{(i)});$

   }
3. *Hash* =Take_n_Least_Significant_Bits$(H^{(N)})$.

General design characteristics for Blue Midnight Wish Hash Function



Definition of the function F_0 of Blue Midnight Wish

$$f_0 : \{0,1\}^{2m} \rightarrow \{0,1\}^m$$

Input: Message block $M^{(i)} = (M_0^{(i)}, M_1^{(i)}, \ldots, M_{15}^{(i)})$, and the previous double pipe $H^{(i-1)} = (H_0^{(i-1)}, H_1^{(i-1)}, \ldots, H_{15}^{(i-1)})$.
Output: First part of the quadruple pipe $Q_a^{(i)} = (Q_0^{(i)}, Q_1^{(i)}, \ldots, Q_{15}^{(i)})$.

1. Bijective transform of $M^{(i)} \oplus H^{(i-1)}$:

$$\begin{aligned}
W_0^{(i)} &= (M_5^{(i)} \oplus H_5^{(i-1)}) - (M_7^{(i)} \oplus H_7^{(i-1)}) + (M_{10}^{(i)} \oplus H_{10}^{(i-1)}) + (M_{13}^{(i)} \oplus H_{13}^{(i-1)}) + (M_{14}^{(i)} \oplus H_{14}^{(i-1)}) \\
W_1^{(i)} &= (M_6^{(i)} \oplus H_6^{(i-1)}) - (M_8^{(i)} \oplus H_8^{(i-1)}) + (M_{11}^{(i)} \oplus H_{11}^{(i-1)}) + (M_{14}^{(i)} \oplus H_{14}^{(i-1)}) - (M_{15}^{(i)} \oplus H_{15}^{(i-1)}) \\
W_2^{(i)} &= (M_0^{(i)} \oplus H_0^{(i-1)}) + (M_7^{(i)} \oplus H_7^{(i-1)}) + (M_9^{(i)} \oplus H_9^{(i-1)}) - (M_{12}^{(i)} \oplus H_{12}^{(i-1)}) + (M_{15}^{(i)} \oplus H_{15}^{(i-1)}) \\
W_3^{(i)} &= (M_0^{(i)} \oplus H_0^{(i-1)}) - (M_1^{(i)} \oplus H_1^{(i-1)}) + (M_8^{(i)} \oplus H_8^{(i-1)}) - (M_{10}^{(i)} \oplus H_{10}^{(i-1)}) + (M_{13}^{(i)} \oplus H_{13}^{(i-1)}) \\
W_4^{(i)} &= (M_1^{(i)} \oplus H_1^{(i-1)}) + (M_2^{(i)} \oplus H_2^{(i-1)}) + (M_9^{(i)} \oplus H_9^{(i-1)}) - (M_{11}^{(i)} \oplus H_{11}^{(i-1)}) - (M_{14}^{(i)} \oplus H_{14}^{(i-1)}) \\
W_5^{(i)} &= (M_3^{(i)} \oplus H_3^{(i-1)}) - (M_2^{(i)} \oplus H_2^{(i-1)}) + (M_{10}^{(i)} \oplus H_{10}^{(i-1)}) - (M_{12}^{(i)} \oplus H_{12}^{(i-1)}) + (M_{15}^{(i)} \oplus H_{15}^{(i-1)}) \\
W_6^{(i)} &= (M_4^{(i)} \oplus H_4^{(i-1)}) - (M_0^{(i)} \oplus H_0^{(i-1)}) - (M_3^{(i)} \oplus H_3^{(i-1)}) - (M_{11}^{(i)} \oplus H_{11}^{(i-1)}) + (M_{13}^{(i)} \oplus H_{13}^{(i-1)}) \\
W_7^{(i)} &= (M_1^{(i)} \oplus H_1^{(i-1)}) - (M_4^{(i)} \oplus H_4^{(i-1)}) - (M_5^{(i)} \oplus H_5^{(i-1)}) - (M_{12}^{(i)} \oplus H_{12}^{(i-1)}) - (M_{14}^{(i)} \oplus H_{14}^{(i-1)}) \\
W_8^{(i)} &= (M_2^{(i)} \oplus H_2^{(i-1)}) - (M_5^{(i)} \oplus H_5^{(i-1)}) - (M_6^{(i)} \oplus H_6^{(i-1)}) + (M_{13}^{(i)} \oplus H_{13}^{(i-1)}) - (M_{15}^{(i)} \oplus H_{15}^{(i-1)}) \\
W_9^{(i)} &= (M_0^{(i)} \oplus H_0^{(i-1)}) - (M_3^{(i)} \oplus H_3^{(i-1)}) + (M_6^{(i)} \oplus H_6^{(i-1)}) - (M_7^{(i)} \oplus H_7^{(i-1)}) + (M_{14}^{(i)} \oplus H_{14}^{(i-1)}) \\
W_{10}^{(i)} &= (M_8^{(i)} \oplus H_8^{(i-1)}) - (M_1^{(i)} \oplus H_1^{(i-1)}) - (M_4^{(i)} \oplus H_4^{(i-1)}) - (M_7^{(i)} \oplus H_7^{(i-1)}) + (M_{15}^{(i)} \oplus H_{15}^{(i-1)}) \\
W_{11}^{(i)} &= (M_8^{(i)} \oplus H_8^{(i-1)}) - (M_0^{(i)} \oplus H_0^{(i-1)}) - (M_2^{(i)} \oplus H_2^{(i-1)}) - (M_5^{(i)} \oplus H_5^{(i-1)}) + (M_9^{(i)} \oplus H_9^{(i-1)}) \\
W_{12}^{(i)} &= (M_1^{(i)} \oplus H_1^{(i-1)}) + (M_3^{(i)} \oplus H_3^{(i-1)}) - (M_6^{(i)} \oplus H_6^{(i-1)}) - (M_9^{(i)} \oplus H_9^{(i-1)}) + (M_{10}^{(i)} \oplus H_{10}^{(i-1)}) \\
W_{13}^{(i)} &= (M_2^{(i)} \oplus H_2^{(i-1)}) + (M_4^{(i)} \oplus H_4^{(i-1)}) + (M_7^{(i)} \oplus H_7^{(i-1)}) + (M_{10}^{(i)} \oplus H_{10}^{(i-1)}) + (M_{11}^{(i)} \oplus H_{11}^{(i-1)}) \\
W_{14}^{(i)} &= (M_3^{(i)} \oplus H_3^{(i-1)}) - (M_5^{(i)} \oplus H_5^{(i-1)}) + (M_8^{(i)} \oplus H_8^{(i-1)}) - (M_{11}^{(i)} \oplus H_{11}^{(i-1)}) - (M_{12}^{(i)} \oplus H_{12}^{(i-1)}) \\
W_{15}^{(i)} &= (M_{12}^{(i)} \oplus H_{12}^{(i-1)}) - (M_4^{(i)} \oplus H_4^{(i-1)}) - (M_6^{(i)} \oplus H_6^{(i-1)}) - (M_9^{(i)} \oplus H_9^{(i-1)}) + (M_{13}^{(i)} \oplus H_{13}^{(i-1)})
\end{aligned}$$

2. Further bijective transform of $W_j^{(i)}, j = 0, \ldots, 15$:

$$\begin{aligned}
Q_0^{(i)} &= s_0(W_0^{(i)}); & Q_1^{(i)} &= s_1(W_1^{(i)}); & Q_2^{(i)} &= s_2(W_2^{(i)}); & Q_3^{(i)} &= s_3(W_3^{(i)}); \\
Q_4^{(i)} &= s_4(W_4^{(i)}); & Q_5^{(i)} &= s_0(W_5^{(i)}); & Q_6^{(i)} &= s_1(W_6^{(i)}); & Q_7^{(i)} &= s_2(W_7^{(i)}); \\
Q_8^{(i)} &= s_3(W_8^{(i)}); & Q_9^{(i)} &= s_4(W_9^{(i)}); & Q_{10}^{(i)} &= s_0(W_{10}^{(i)}); & Q_{11}^{(i)} &= s_1(W_{11}^{(i)}); \\
Q_{12}^{(i)} &= s_2(W_{12}^{(i)}); & Q_{13}^{(i)} &= s_3(W_{13}^{(i)}); & Q_{14}^{(i)} &= s_4(W_{14}^{(i)}); & Q_{15}^{(i)} &= s_0(W_{15}^{(i)});
\end{aligned}$$

NTNU
Innovation and Creativity

Definition of the function F_1 of Blue Midnight Wish

For ii = 0,1 : $Q^{(i)}_{(ii+16)} = Expand_1(ii + 16)$

$$Expand_1(i) = S_1(Q^i_{(j-16)}) + S_2(Q^i_{(j-15)}) + S_3(Q^i_{(j-14)}) + S_0(Q^i_{(j-13)}) + S_1(Q^i_{(j-12)}) + S_2(Q^i_{(j-11)}) + S_3(Q^i_{(j-10)}) + S_0(Q^i_{(j-9)}) + S_1(Q^i_{(j-8)}) + S_2(Q^i_{(j-7)}) + S_3(Q^i_{(j-6)}) + S_0(Q^i_{(j-5)}) + S_1(Q^i_{(j-4)}) + S_2(Q^i_{(j-3)}) + S_3(Q^i_{(j-2)}) + S_0(Q^i_{(j-1)}) + ADD\_Element(j - 16)$$

For ii=2,3,4,5,...,15 : $Q^{(i)}_{(ii+16)} = Expand_2(ii + 16)$

$$Expand_2(i) = (Q^i_{(j-16)}) + r_1(Q^i_{(j-15)}) + (Q^i_{(j-14)}) + r_2(Q^i_{(j-13)}) + (Q^i_{(j-12)}) + r_3(Q^i_{(j-11)}) + (Q^i_{(j-10)}) + r_4(Q^i_{(j-9)}) + (Q^i_{(j-8)}) + r_5(Q^i_{(j-7)}) + (Q^i_{(j-6)}) + r_6(Q^i_{(j-5)}) + (Q^i_{(j-4)}) + r_7(Q^i_{(j-3)}) + S_4(Q^i_{(j-2)}) + S_5(Q^i_{(j-1)}) + ADD\_Element(j - 16)$$

$$ADD\_Element(j) = (ROTL^{(j+1)}(M^{(i)}_{(j)}) + ROTL^{(j+4)}(M^{(i)}_{(j+3)}) + ROTL^{(j+11)}(M^{(i)}_{(j+10)}) + K_{j+16}) \oplus H^{(i)}_{j+7}$$

NTNU
Innovation and Creativity

Definition of the function F_2 of Blue Midnight Wish

Folding $f_2 : \{0,1\}^{3m} \rightarrow \{0,1\}^m$

Input: Message block $M^{(i)} = (M_0^{(i)}, M_1^{(i)}, \ldots, M_{15}^{(i)})$,
quadruple pipe $Q^{(i)} = (Q_0^{(i)}, Q_1^{(i)}, \ldots, Q_{15}^{(i)}, Q_{16}^{(i)}, \ldots, Q_{31}^{(i)})$.
Output: New double pipe $H^{(i)} = (H_0^{(i)}, H_1^{(i)}, \ldots, H_{15}^{(i)})$.

1. Compute the cumulative temporary variables XL and XH.

$$XL = Q_{16}^{(i)} \oplus Q_{17}^{(i)} \oplus \ldots \oplus Q_{23}^{(i)}$$
$$XH = XL \oplus Q_{24}^{(i)} \oplus Q_{25}^{(i)} \oplus \ldots \oplus Q_{31}^{(i)}$$

2. Compute the new double pipe $H^{(i)}$:

$$\begin{aligned}
H_0^{(i)} &= \left(SHL^5(XH) \oplus SHR^5(Q_{16}^{(i)}) \oplus M_0^{(i)}\right) + \left(XL \oplus Q_{24}^{(i)} \oplus Q_0^{(i)}\right)\\
H_1^{(i)} &= \left(SHR^7(XH) \oplus SHL^8(Q_{17}^{(i)}) \oplus M_1^{(i)}\right) + \left(XL \oplus Q_{25}^{(i)} \oplus Q_1^{(i)}\right)\\
H_2^{(i)} &= \left(SHR^5(XH) \oplus SHL^5(Q_{18}^{(i)}) \oplus M_2^{(i)}\right) + \left(XL \oplus Q_{26}^{(i)} \oplus Q_2^{(i)}\right)\\
H_3^{(i)} &= \left(SHR^1(XH) \oplus SHL^5(Q_{19}^{(i)}) \oplus M_3^{(i)}\right) + \left(XL \oplus Q_{27}^{(i)} \oplus Q_3^{(i)}\right)\\
H_4^{(i)} &= \left(SHR^3(XH) \oplus Q_{20}^{(i)} \oplus M_4^{(i)}\right) + \left(XL \oplus Q_{28}^{(i)} \oplus Q_4^{(i)}\right)\\
H_5^{(i)} &= \left(SHL^6(XH) \oplus SHR^6(Q_{21}^{(i)}) \oplus M_5^{(i)}\right) + \left(XL \oplus Q_{29}^{(i)} \oplus Q_5^{(i)}\right)\\
H_6^{(i)} &= \left(SHR^4(XH) \oplus SHL^6(Q_{22}^{(i)}) \oplus M_6^{(i)}\right) + \left(XL \oplus Q_{30}^{(i)} \oplus Q_6^{(i)}\right)\\
H_7^{(i)} &= \left(SHR^{11}(XH) \oplus SHL^2(Q_{23}^{(i)}) \oplus M_7^{(i)}\right) + \left(XL \oplus Q_{31}^{(i)} \oplus Q_7^{(i)}\right)\\
H_8^{(i)} &= ROTL^9(H_4^{(i)}) + \left(XH \oplus Q_{24}^{(i)} \oplus M_8^{(i)}\right) + \left(SHL^8(XL) \oplus Q_{23}^{(i)} \oplus Q_8^{(i)}\right)\\
H_9^{(i)} &= ROTL^{10}(H_5^{(i)}) + \left(XH \oplus Q_{25}^{(i)} \oplus M_9^{(i)}\right) + \left(SHR^6(XL) \oplus Q_{16}^{(i)} \oplus Q_9^{(i)}\right)\\
H_{10}^{(i)} &= ROTL^{11}(H_6^{(i)}) + \left(XH \oplus Q_{26}^{(i)} \oplus M_{10}^{(i)}\right) + \left(SHL^6(XL) \oplus Q_{17}^{(i)} \oplus Q_{10}^{(i)}\right)\\
H_{11}^{(i)} &= ROTL^{12}(H_7^{(i)}) + \left(XH \oplus Q_{27}^{(i)} \oplus M_{11}^{(i)}\right) + \left(SHL^4(XL) \oplus Q_{18}^{(i)} \oplus Q_{11}^{(i)}\right)\\
H_{12}^{(i)} &= ROTL^{13}(H_0^{(i)}) + \left(XH \oplus Q_{28}^{(i)} \oplus M_{12}^{(i)}\right) + \left(SHR^3(XL) \oplus Q_{19}^{(i)} \oplus Q_{12}^{(i)}\right)\\
H_{13}^{(i)} &= ROTL^{14}(H_1^{(i)}) + \left(XH \oplus Q_{29}^{(i)} \oplus M_{13}^{(i)}\right) + \left(SHR^4(XL) \oplus Q_{20}^{(i)} \oplus Q_{13}^{(i)}\right)\\
H_{14}^{(i)} &= ROTL^{15}(H_2^{(i)}) + \left(XH \oplus Q_{30}^{(i)} \oplus M_{14}^{(i)}\right) + \left(SHR^7(XL) \oplus Q_{21}^{(i)} \oplus Q_{14}^{(i)}\right)\\
H_{15}^{(i)} &= ROTL^{16}(H_3^{(i)}) + \left(XH \oplus Q_{31}^{(i)} \oplus M_{15}^{(i)}\right) + \left(SHR^2(XL) \oplus Q_{22}^{(i)} \oplus Q_{15}^{(i)}\right)
\end{aligned}$$

NTNU
Innovation and Creativity

Definition of Logic Operations of Blue Midnight Wish

3. S-transform used in f_0 Function
$S_0(x) = SHR^1(x) \oplus SHL^3(x) \oplus ROTL^4(x) \oplus ROTL^{19}(x)$
$S_1(x) = SHR^1(x) \oplus SHL^2(x) \oplus ROTL^8(x) \oplus ROTL^{23}(x)$
$S_2(x) = SHR^2(x) \oplus SHL^1(x) \oplus ROTL^{12}(x) \oplus ROTL^{25}(x)$
$S_3(x) = SHR^2(x) \oplus SHL^2(x) \oplus ROTL^{15}(x) \oplus ROTL^{29}(x)$

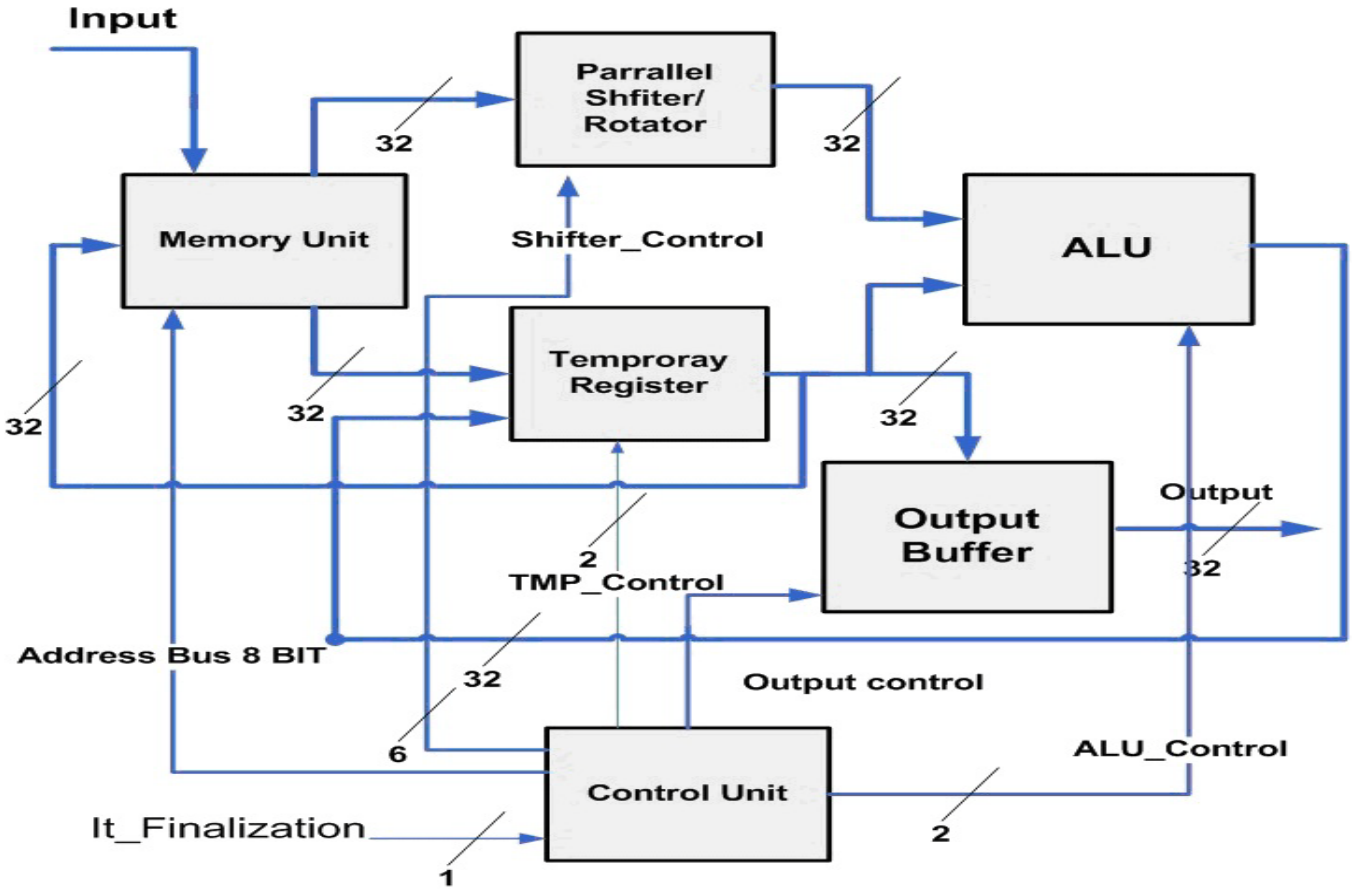
R- transform

$$r_1(x) = ROTL^3(X), r_2(x) = ROTL_7(x),$$
$$r_3(x) = ROTL^{13}(x), r_4(x) = ROTL^{16}(x),$$
$$r_5(x) = ROTL^{19}(x), r_6(x) = ROTL^{23}(x),$$
$$r_7(x) = ROTL^{27}(x), S_4(x) = SHR^1(x) \oplus x,$$
$$S_5(x) = SHR^2(x) \oplus x$$

Outline

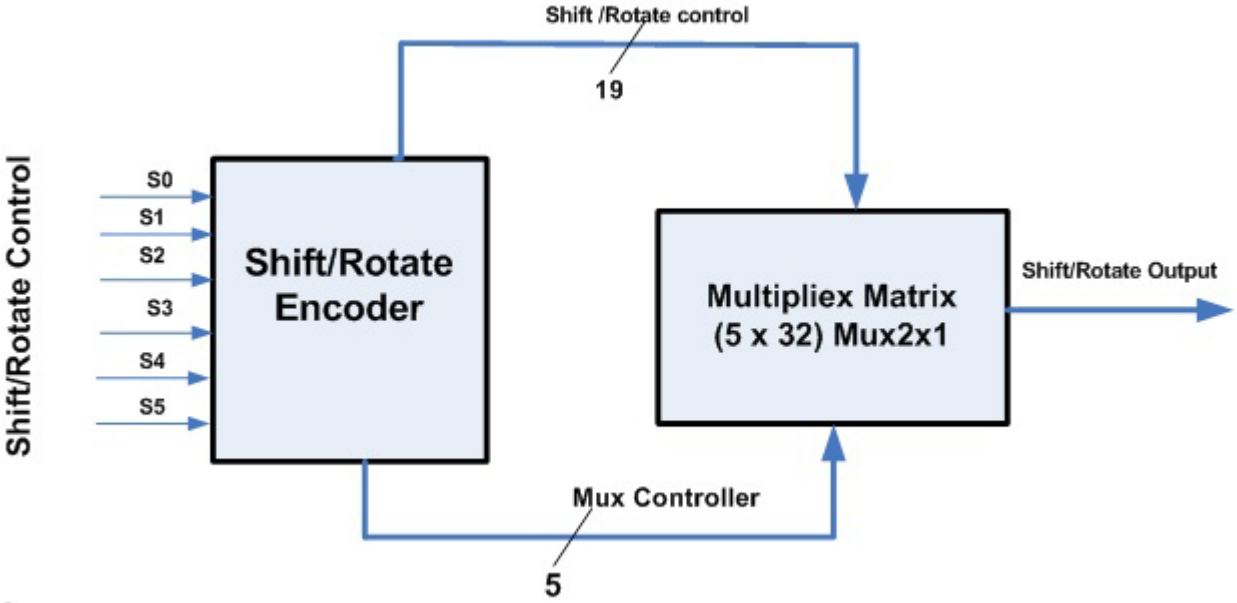
- General characteristics of **B**lue **M**idnight **W**ish
- **BMW-256 Hashing Core FPGA Architecture**
- BMW-256 Hashing Core Operations
- FPGA Implementations Characteristics
- Conclusions

BMW-256 Hashing Core Architecture



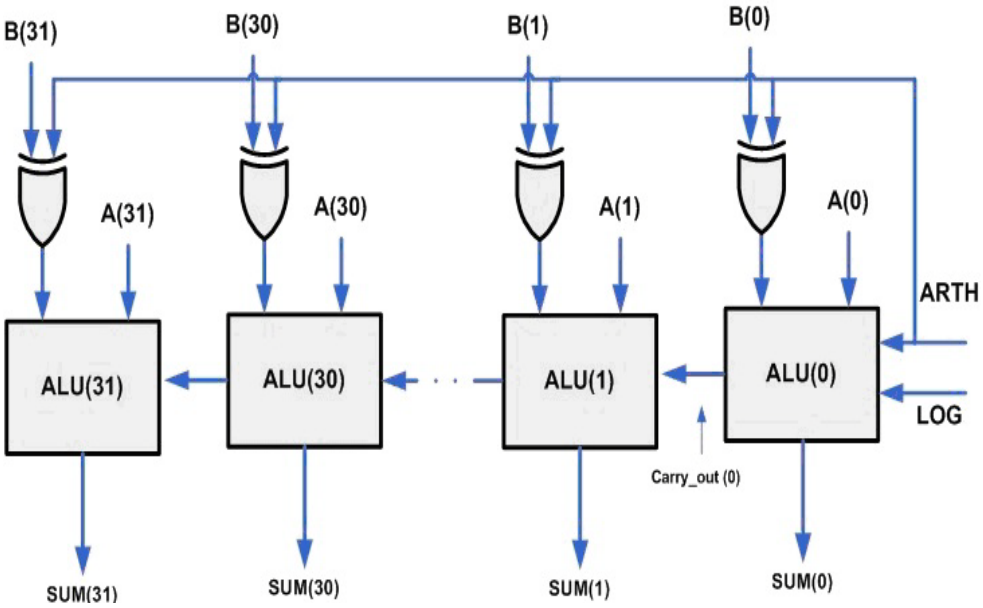
the complete architecture of the entire BMW core process, which includes six main hardware operative parts, Memory units, Parallel shifter/Rotator, ALU (Arithmetic logic unit), Temporary Register, Output stage and Control Unit

Parallel Shifter/Rotator: It contains a 5 x 32 Mux (Multiplex) matrix each one is Mux 2x1 with big Encoder (5 X 11). This component is responsible for the shift and rotation operations of the 32 bit word. It receives 32 bit parallel data from the Memory Block and transmits 32 bit parallel data to the ALU. That happens decided by the value of shifter control word. Because we have 46 operations in BMW hash Core, the width of shifter control word is 6 control bits

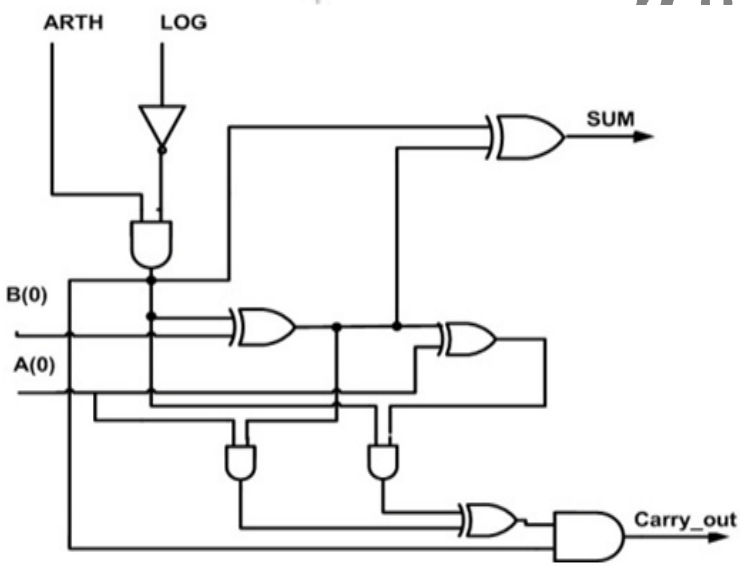


Parallel Shifter/Rotator operations in BLUE MIDNIGHT WISH-256 Core

S/R control	Operation	S/R control	Operation
000000	LOAD	010111	ROL8
000001	SHl1	011000	ROL9
000010	SHl2	011001	ROL10
000011	SHl3	011010	ROL11
000100	SHl4	011011	ROL12
000101	SHl5	011100	ROL13
000110	SHl6	011101	ROL14
000111	SHl8	011110	ROL15
001000	SHR1	011111	ROL16
001001	SHR2	100000	ROL17
001010	SHR3	100001	ROL18
001011	SHR4	100010	ROL19
001100	SHR5	100011	ROL20
001101	SHR6	100100	ROL21
001110	SHR7	100101	ROL22
001111	SHR11	100110	ROL23
010000	ROL1	100111	ROL24
010001	ROL2	101000	ROL25
010010	ROL3	101001	ROL26
010011	ROL4	101010	ROL27
010100	ROL5	101011	ROL28
010101	ROL6	101100	ROL29
010110	ROL7	101101	ROL30

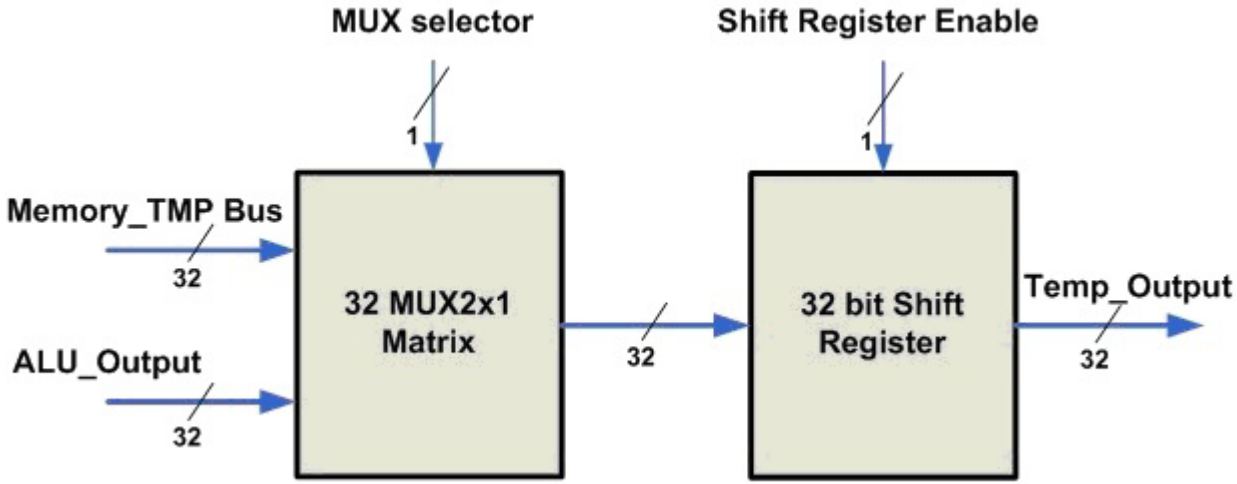


Parallel 32 bit ALU (Arithmetic and Logic Unit)

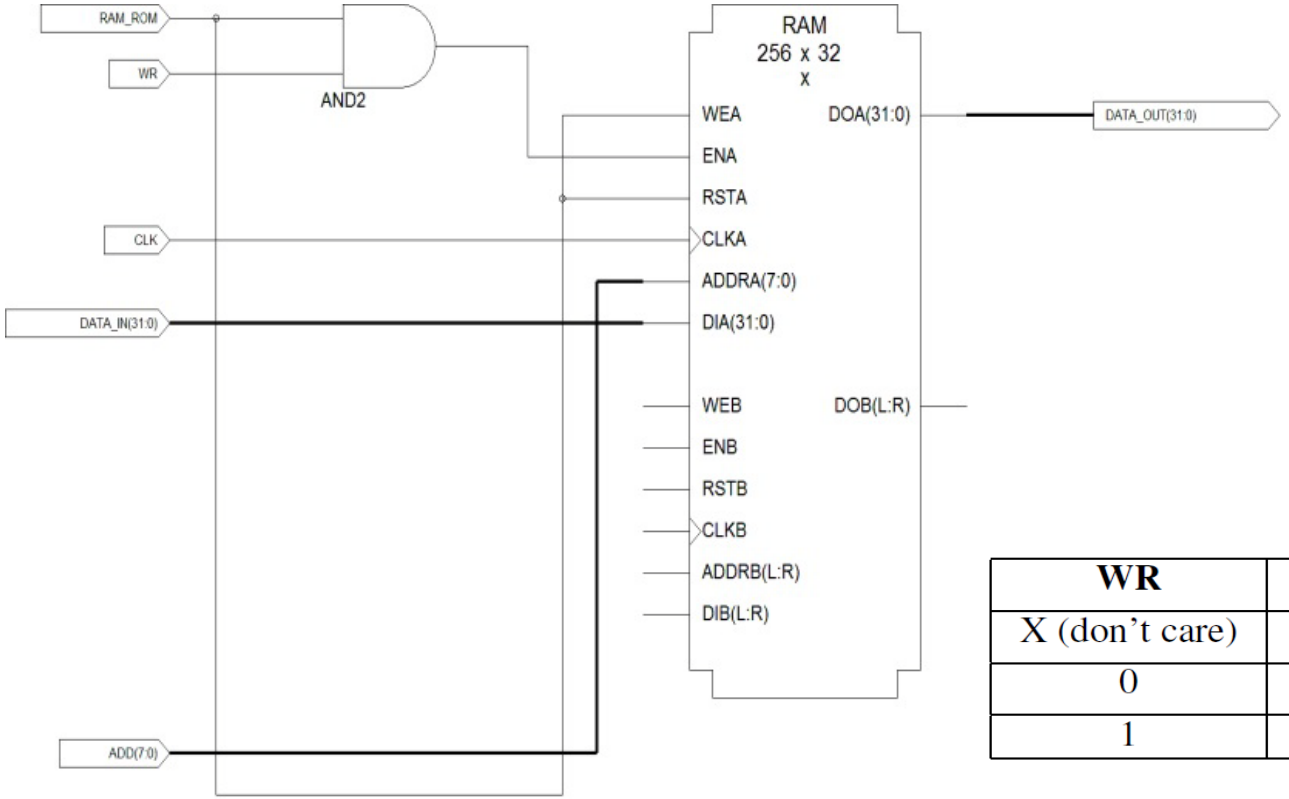


ALU Cell (Arithmetic and Logic Unit)

ALU component: The ALU component offers four different operations in the hash computation stage: bit-wise logical word XOR operation, word addition and subtraction (modulo 2^{32}). The ALU component receives 32 bit data words from the parallel shifter/rotator and the Temporary Register and transmit the output to the Temporary Register to work as a parallel accumulator.

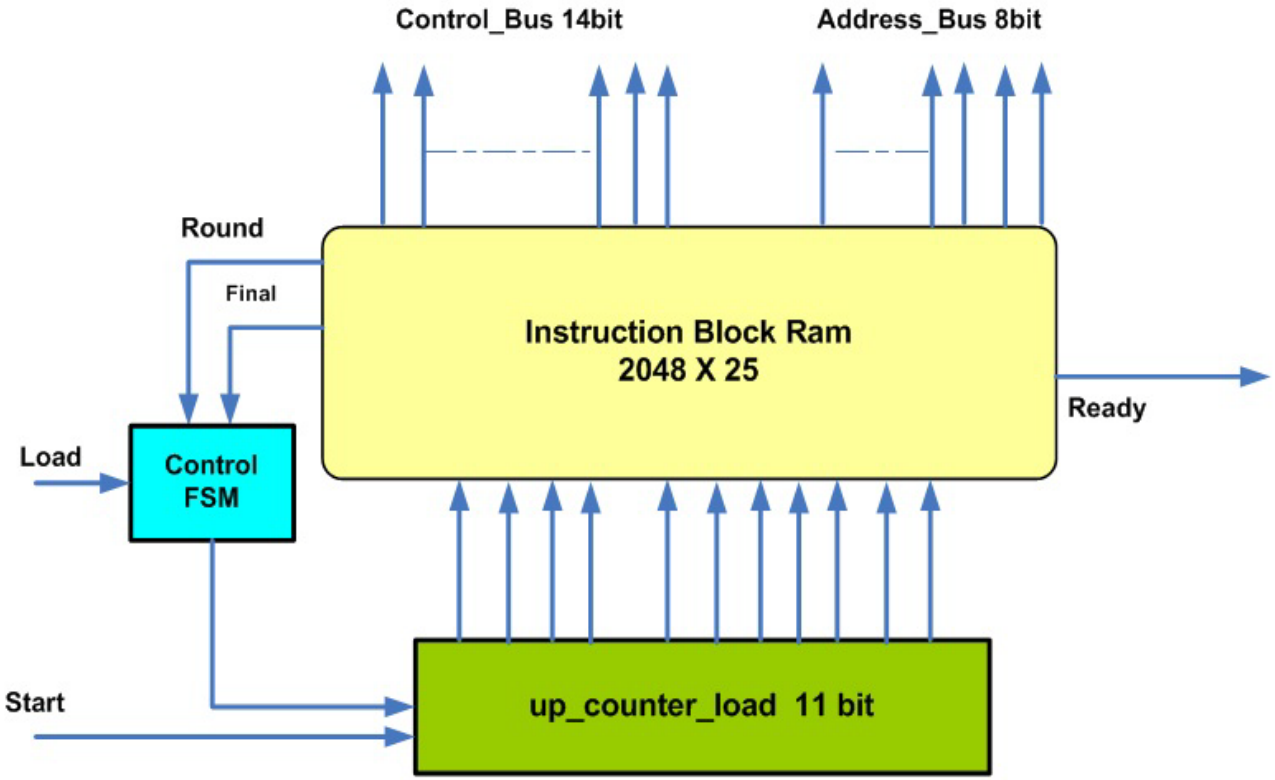


Temporary Register: It contains 32 Mux 2x1 and a shift register. The Temporary Register works as an accumulator. It receives 32 bit words from The Memory unit and The ALU and transmits data 32 bit words to The ALU and the output stage.



WR	RAM_ROM	Operation
X (don't care)	1	ROM (Read)
0	0	RAM (Read)
1	0	RAM (Write)

Memory Block: To implement the BMW-256 Core Memory block, we used an FPGA block RAM of size 256 x 32 bits. The Memory Block contains ROM to store the BMW-256 constants. In addition, the Memory Block contains RAM to store the BMW-256 input block Message ($M^{(i)}_0$,$M^{(i)}_1$,..., $M^{(i)}_{15}$), the intermediate values of the BMW hash function, and the final double pipe values

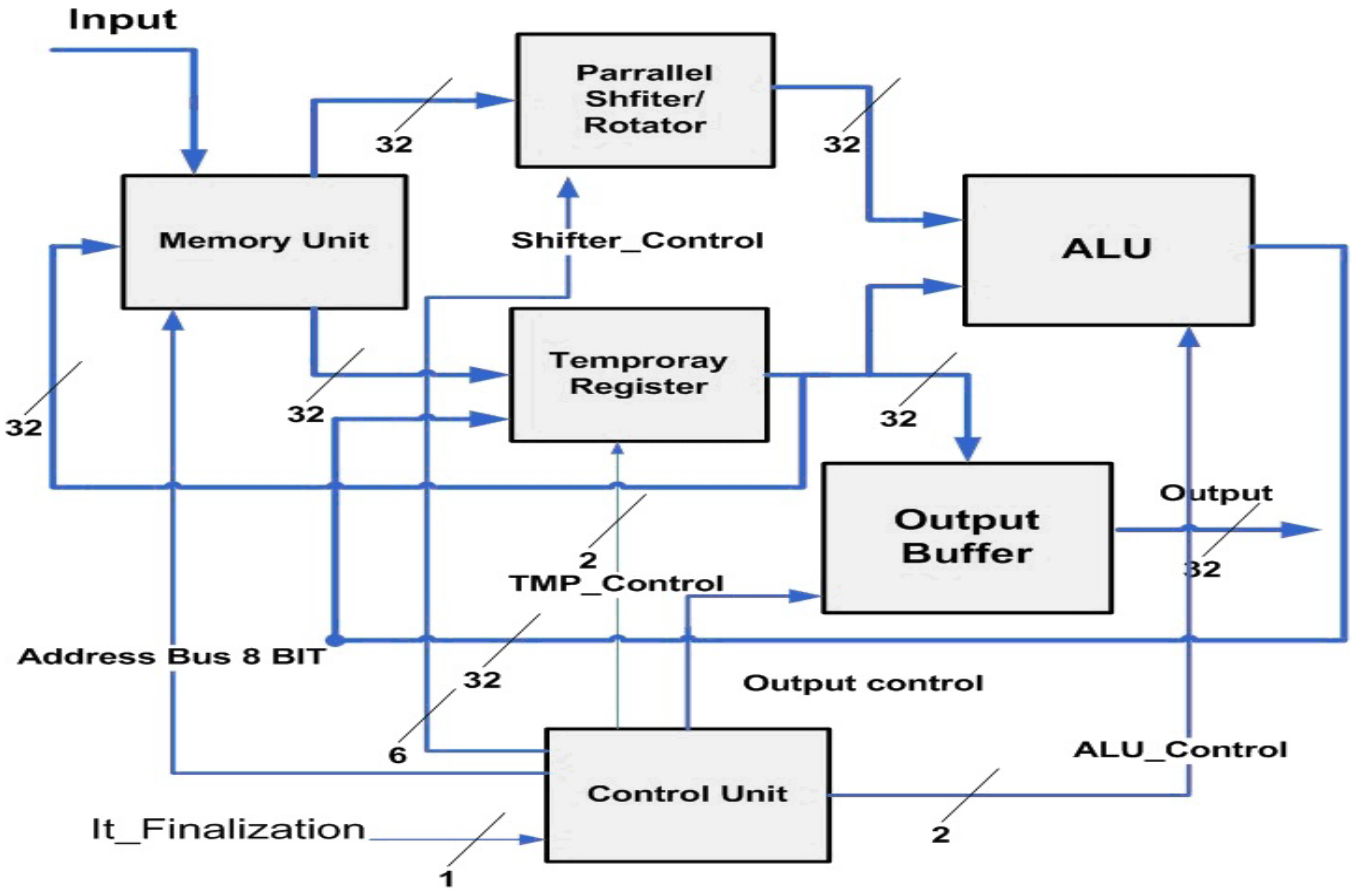


Control Unit: Control Unit: It has been designed as a 2048 x 25 bit Instruction Block RAM, an 11 bit up counter load bit and a Control FSM (Finite State Machine). it contains three operative parts, all of them working together to produce 8 bit memory address words to control the memory block traffic with the other BMW-256 sub-systems. The Control Unit produces the 14 bit control word to control the data flow between the BMW-256 core sub-systems.

Outline

- General characteristics of **B**lue **M**idnight **W**ish
- BMW-256 Hashing Core FPGA Architecture
- **BMW-256 Hashing Core Operations**
- FPGA Implementations Characteristics
- Conclusions

BMW-256 Hashing Core Operation



BLUE MIDNIGHT WISH-256 HASHING CORE OPERATIONS (EXECUTION TIMES)

Operation	Proposed	BMW[*]	Operation	Proposed	BMW [*]
XOR	3	32	S_5	2	34
ADD	1	32	R_1	1	3
SUB	1	32	R_2	1	7
S_0	4	127	R_3	1	13
S_1	4	128	R_4	1	16
S_2	4	129	R_5	1	19
S_3	4	132	R_6	1	23
S_4	4	34	R_7	1	27

[*] M. El Hadedy, D. Gligoroski, S. J. Knapskog, ”Low Area Implementation of the Hash Function ”Blue Midnight Wish - 256” for FPGA platforms”. In Proceedings of The International Conference on Intelligent Networking and Collaborative Systems. IEEE Computer Society 2009 ISBN 978-0-7695-3858-7.

BLUE MIDNIGHT WISH internal functions (execution times)

BMW functions	No. of Cycles
F_0	413
F_1	476
F_2	171

Outline

- General characteristics of **B**lue **M**idnight **W**ish
- BMW-256 Hashing Core FPGA Architecture
- BMW-256 Hashing Core Operations
- **FPGA Implementations Characteristics**
- Conclusions

FPGA Implementations Characteristics

Algorithm Name	FPGA Implementation				
	FPGA Type	Area		Frequency	Estimated Throughput
		Slices	BRAM		
Proposed	Virtex	895	1	38 MHz	9 Mbps
	Virtex 5	84	2	116 MHz	28 Mbps

Outline

- General characteristics of **B**lue **M**idnight **W**ish
- BMW-256 Hashing Core FPGA Architecture
- BMW-256 Hashing Core Operations
- FPGA Implementations Characteristics
- **Conclusions**

Conclusions

- We have presented an FPGA implementation of BMW-256 hash function
- It produces 256 bits of message digest using a parallel shifter/rotator and a parallel 32 bit word arithmetic logic unit (ALU).
- The BMW-256 core receives 16 message words of 32 bits and processes them.
- The present solution uses as small area (slice) as possible in order to minimize the hardware cost.
- In our future work, we will take on the challenge to improve the throughput while keeping the optimized the area usage.
- In future usage scenarios, full implementation in ASIC can be an option .

My ongoing newest implementations

(not in the NIST paper)

Algorithm Name	FPGA Implementation				
	Device Name	Area		Frequency	Estimated Throughput
		Slice	Block RAM		
BMW 256	XC5VLX110	60	2	116 MHz	28 Mbps
BMW 512	XC5VLX110	118	2	62 MHz	16 Mbps

Thank you for listening!