Implementation of SM4 Block cipher on CUDA GPU and its analysis

Si-Woo Eum, Hyun-Jun Kim, Hyeok-Dong Kwon,
Kyung-Bae Jang, Hyun-Ji Kim, Hwa-Jeong Seo*

한성대학교 HANSUNG UNIVERSITY

CryptoCraft LAB

1. Introduction

- SM4 Block cipher is a symmetric key algorithm developed by the China National Cryptographic Authority.
  - Its simple design can be applied to various smart devices.

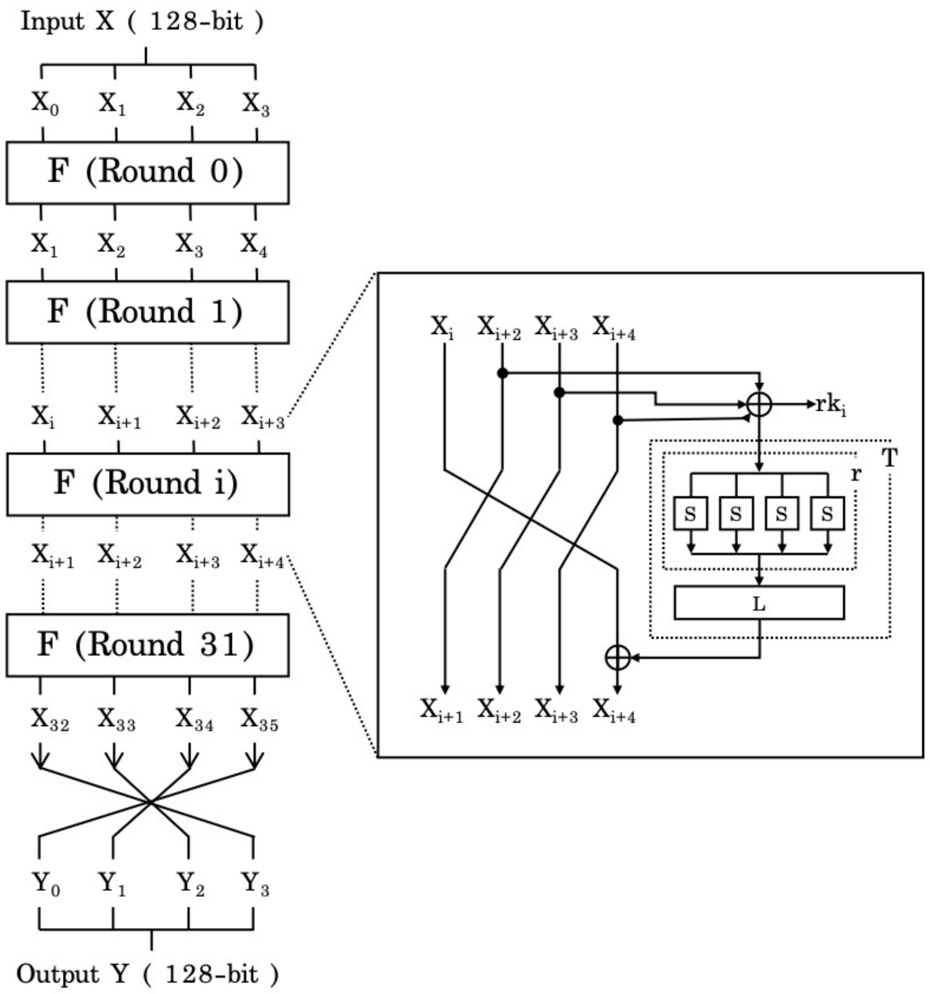
- Graphics Processing Unit(GPU) have become an integral part of today's major computing systems.
  - Various studies using GPU for parallel implementation of block ciphers are also being conducted.

- We implement SM4 Block cipher on GPU and its analysis.

2. Related Work : SM4 Block Cipher

- SM4 Block cipher Parameter
  - Block size : 128-bit
  - Key size : 256-bit
  - Rounds : 32 rounds

- Round Function
  - T function : consists of tau and L.
  - tau function : substituted through Sbox.
  - L function : a linear substitution function



2. Related Work : SM4 Block Cipher

- T function

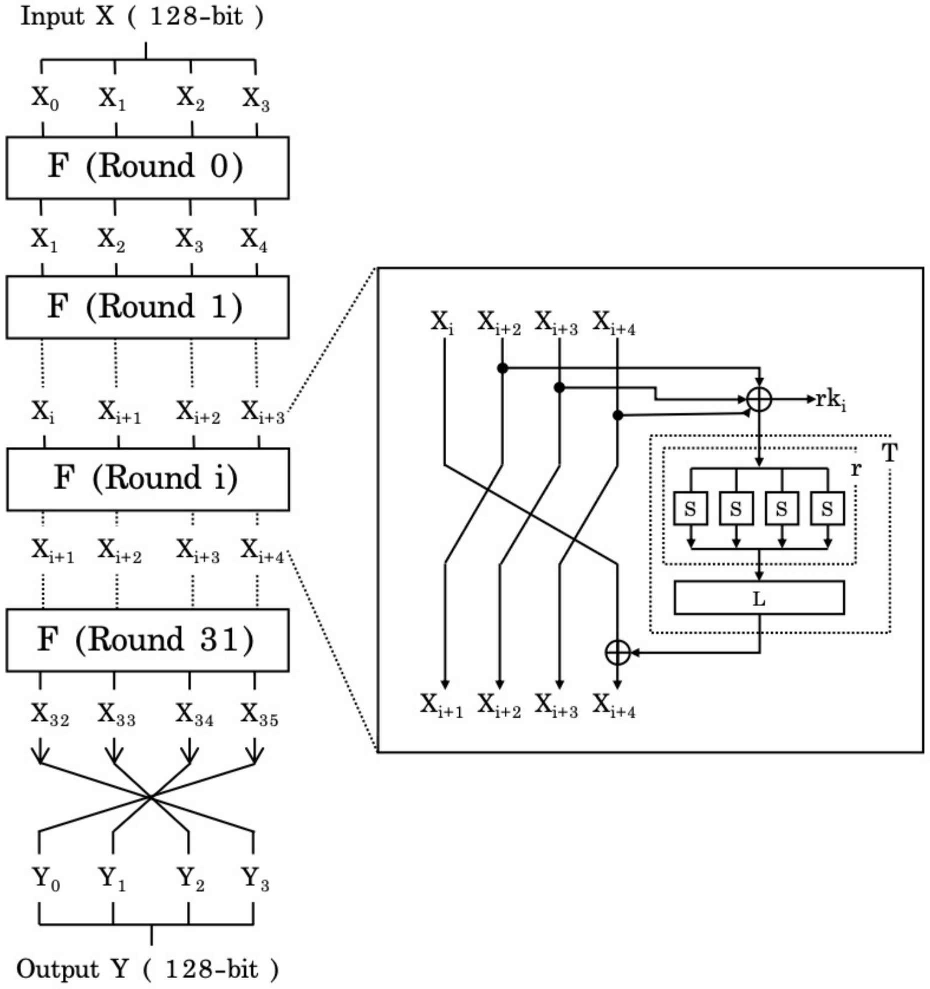
$$\mathtt{T}(\mathbf{A}) = \mathtt{L}(\mathtt{tau}(\mathbf{A})).$$

- tau function

$$\mathtt{tau}(\mathbf{A}) = (\mathrm{Sbox}(a_0), \mathrm{Sbox}(a_1), \mathrm{Sbox}(a_2), \mathrm{Sbox}(a_3)) = (\mathrm{b}_0, \mathrm{b}_1, \mathrm{b}_2, \mathrm{b}_3)$$

- L function

$$\mathbf{C} = \mathtt{L}(\mathbf{B}) = B \oplus (B \lll 2) \oplus (B \lll 10) \oplus (B \lll 18) \oplus (B \lll 24)$$



2. Related Work : GPU

- GPU is parallel programmable processor that feature arithmetic and memory bandwidths that exceed CPU.

- CUDA (Computing Unified Device Architecture) is a parallel computi ng platform developed by Nvidia. CUDA comes with a software en vironment that allows developers to use C as a high-level program ming language.

- The CUDA GPU architecture includes Thread, Block, Grid, Warp, an d Functional kernel running on the GPU

2. Related Work : GPU

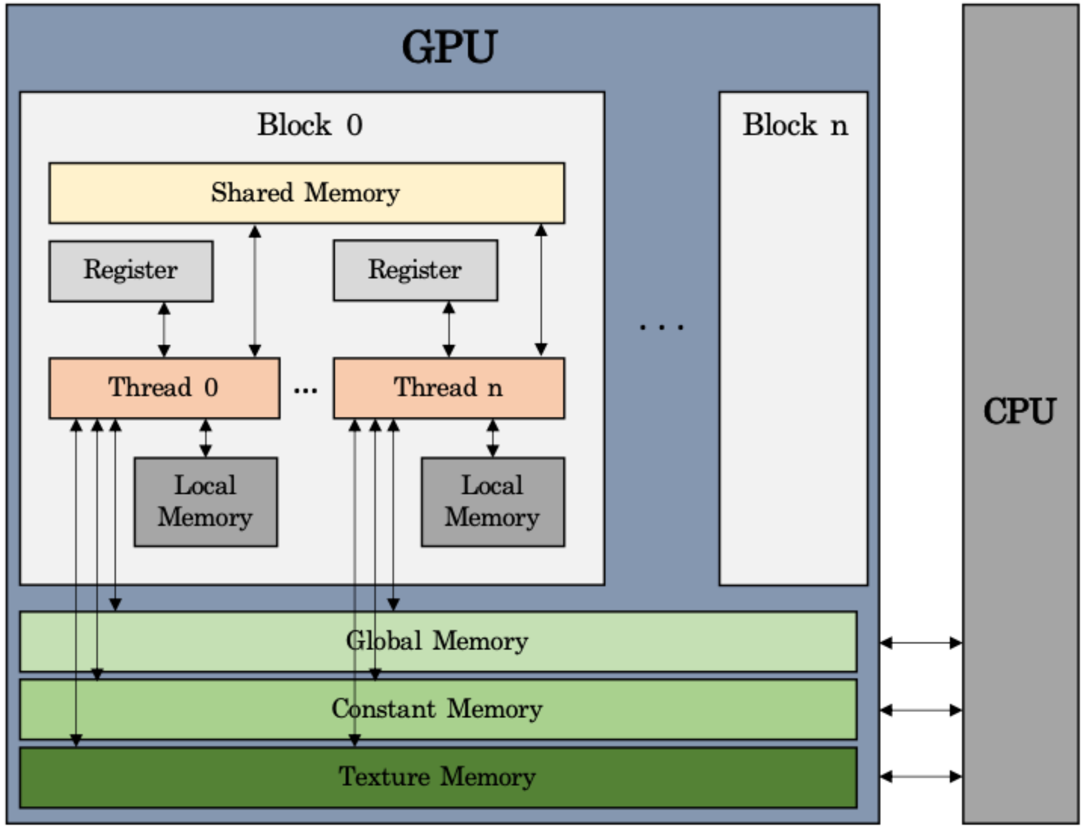
- GPU provide different types of memory

  - Global memory
  - Constant memory
  - Texture memory
  - Local memory

Off-chip

  - Register
  - Shared memory

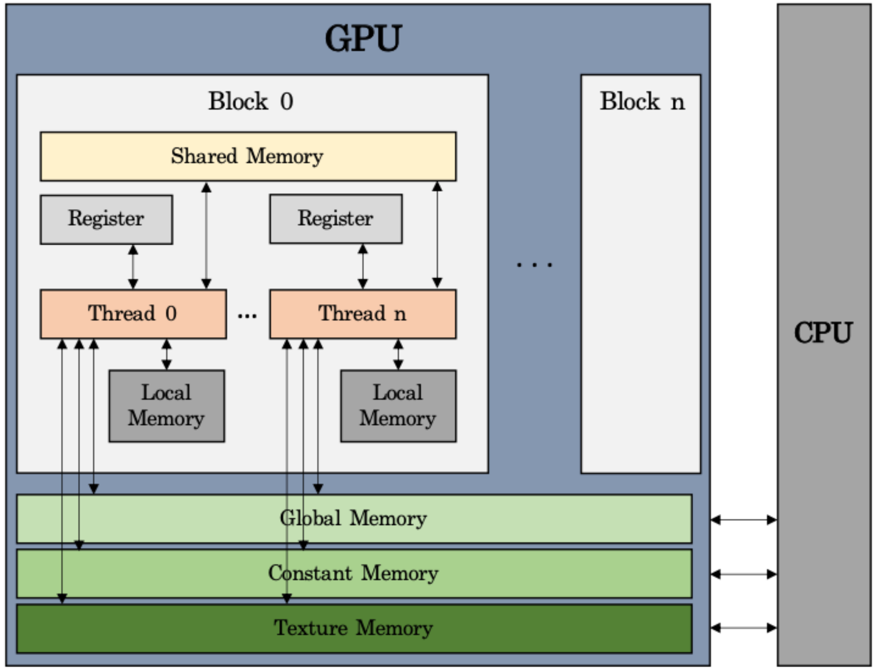
On-chip



2. Related Work : GPU

- Memory Access Speed
  - Register 〉 Shared Memory 〉 Global Memory

- Memory Capacity
  - Global Memory 〉 Shared Memory 〉 Register

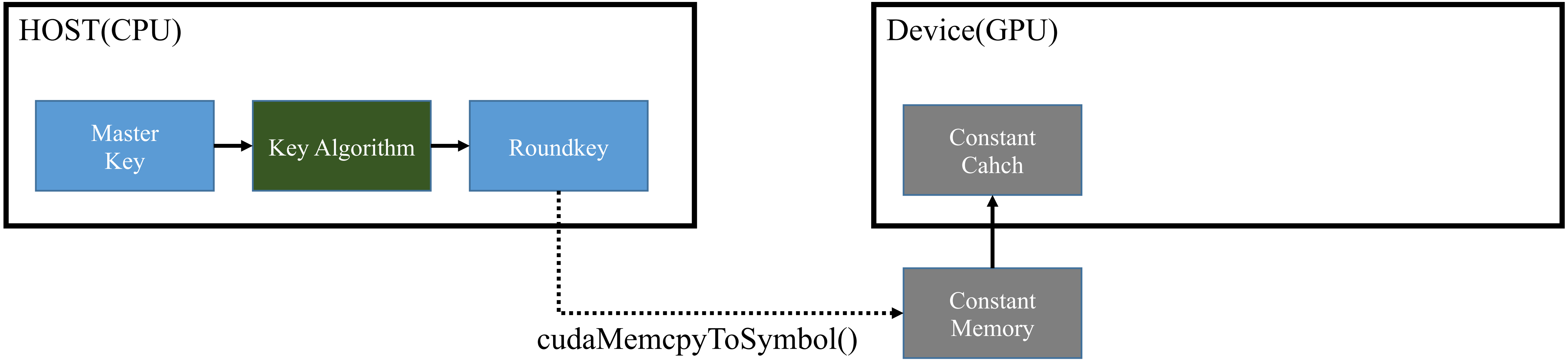


3. Implementation on GPU : SM4 T-table

- SM4 is implemented using 8-bit sbox table(S-table) and 32-bit T-table.
  - The size of the GPU registers is 32-bit, so using a T-table is expected to give better performance.
- The difference between the two implementations is that table size and implementation of T function.
  - S-table size is 256 Byte. - 1byte * 256
  - T-table size is 4,096 Byte(4KB). - 4byte * 256 * 4 tables
- The implementation of the T function is simplified because the L function is precomputed in the T-table implementation.

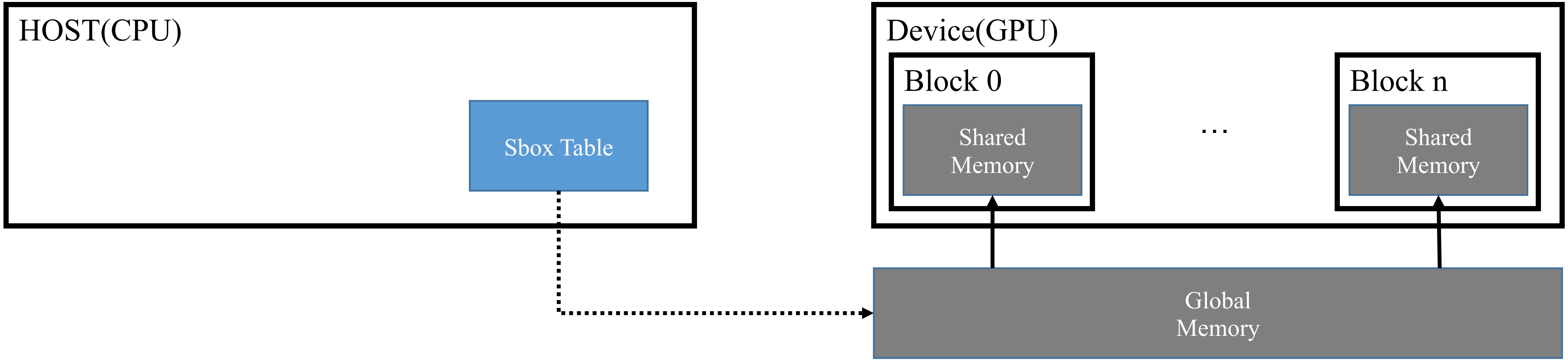
3. Implementation on GPU : Constant memory

- Constant memory is a read only memory.
  - Kernel can only read values from constant memory. It is initialized in the Host.
- If all threads use values stored at the same address, Constant memory access speed is fast much as shared memory.



3. Implementation on GPU : Shared Memory

- Shared Memory has High bandwidth and low latency.
- In *tau* Function, it is calculated through many memory accesses.




```
__shared__ uint8_t S[256];

for (i = threadIdx.x; i < 256; i+=blockDim.x)
{
    S[i] = S_t[i];
}
```

4. Conclusion : Performance comparison

TABLE I

PERFORMANCE OF SM4 BLOCK CIHPER USING 8-BIT SBOX STORED IN GLOBAL MEMORY(UNIT: KB/S(KILO BLOCK ENCRYPTION PER SECOND))

SM4_Global S-table		Threads					
		32	64	128	256	512	1024
Girds	1024 * 8	2746781.25	3011764.75	**3097398.5**	3031379.5	2626483	1606936.25
	1024 * 16	2833425.5	3068624.5	**3119573.5**	3057629.25	2659999.5	1605959.5
	1024 * 32	2878830.5	3097867.25	**3127195.25**	3070580	2674022.25	1972573.125
	1024 * 64	2900439	**3111752.75**	3077269.75	3077269.75	2812197	2063320.25
	1024 * 128	2912607.5	3119811.25	**3151375.5**	3082538.5	2972410.25	2040644.125

TABLE II

PERFORMANCE OF SM4 BLOCK CIHPER USING 8-BIT SBOX STORED IN SHARED MEMORY(UNIT: KB/S(KILO BLOCK ENCRYPTION PER SECOND))

SM4_Shared S-table		Threads					
		32	64	128	256	512	1024
Girds	1024 * 8	2797814.25	3196004.75	**3295784**	3190031.25	2915095	1585476.75
	1024 * 16	2910744.75	3253892.5	**3324675.5**	3255961.75	3064606.75	1605298.75
	1024 * 32	2964678.75	3279423.5	**3339856.5**	3307627	3160860	1731146.125
	1024 * 64	2990217.5	3292604.75	**3346678.75**	3334826	3204757.25	2056502.25
	1024 * 128	3003153	3299633.5	3350374.25	3525905	**3539520.75**	2076374.875

TABLE III

PERFORMANCE OF SM4 BLOCK CIHPER USING 32-BIT T-TABLE STORED IN GLOBAL MEMORY(UNIT: KB/S(KILO BLOCK ENCRYPTION PER SECOND))

SM4_Global T-table		Threads					
		32	64	128	256	512	1024
Girds	1024 * 8	2267493.5	2323049	**2348623.75**	2288268.25	1910804.125	1396331.875
	1024 * 16	2321995.5	2347008.75	**2362168.25**	2312817.75	1942152.75	1379949.25
	1024 * 32	2117890.5	**2358903.25**	2060154.875	2326677.75	1934036.875	1721767.875
	1024 * 64	2189203.75	2367356.25	**2372979.5**	2286001.25	2145358.75	1804748.25
	1024 * 128	2275176.5	2278910.75	2327537.5	**2342964.75**	2333570.5	1805061.125

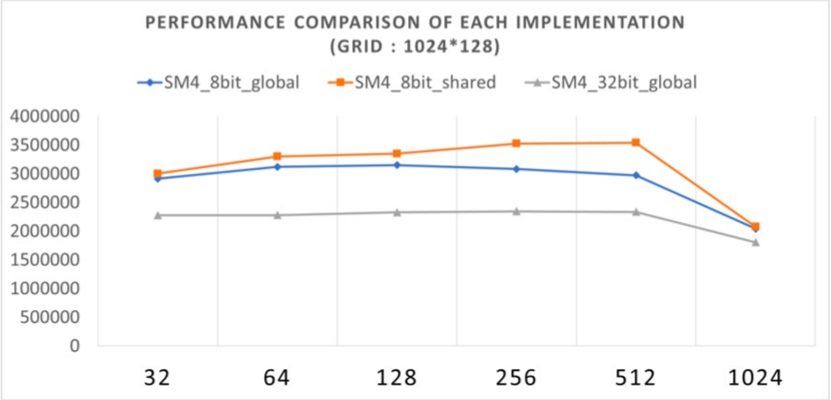


Fig. 3. Comparison of performance measures for each implementation(Grid size : 1024*128).

4. Conclusion : Summary

- We implement SM4 block cipher on GPU
- In this paper, the performance of the implementation using the S-table and using the t-table is presented for comparison. And also, the performance of the implementation using the global memory and using the shared memory is presented for comparison.
- We hope that this study will help other block cipher implementations on GPUs.

Thank you

shuraatum@gmail.com