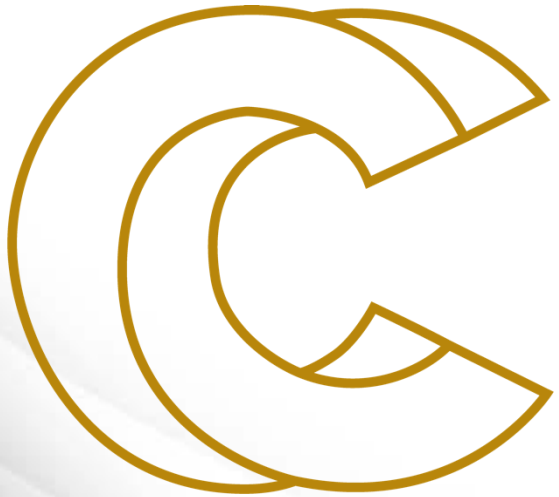


EURO^2

GPU Optimization of Advanced Encryption Standard

Cihangir Tezcan, PhD

Graduate School of Informatics, METU, Ankara

ncc@ulakbim.gov.tr

Lesson 4: Reference C and CUDA Implementation of AES

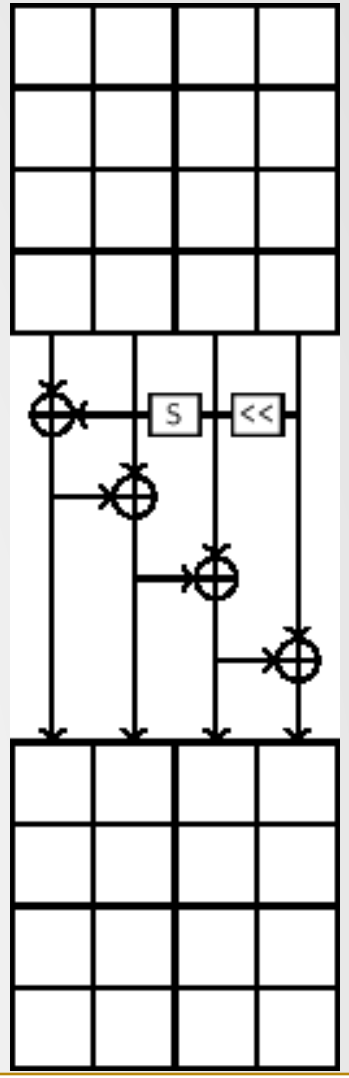
Type Definitions

```
typedef unsigned char       u8;
typedef unsigned short      u16;
typedef unsigned int        u32;
typedef unsigned long long  u64;


u32 RCON32[RCON_SIZE] = {
        0x01000000, 0x02000000, 0x04000000, 0x08000000,
        0x10000000, 0x20000000, 0x40000000, 0x80000000,
        0x1B000000, 0x36000000, 0x6C000000, 0xD8000000,
        0xAB000000, 0x4D000000, 0x9A000000
};
```

Lesson 4: Reference C and CUDA Implementation of AES

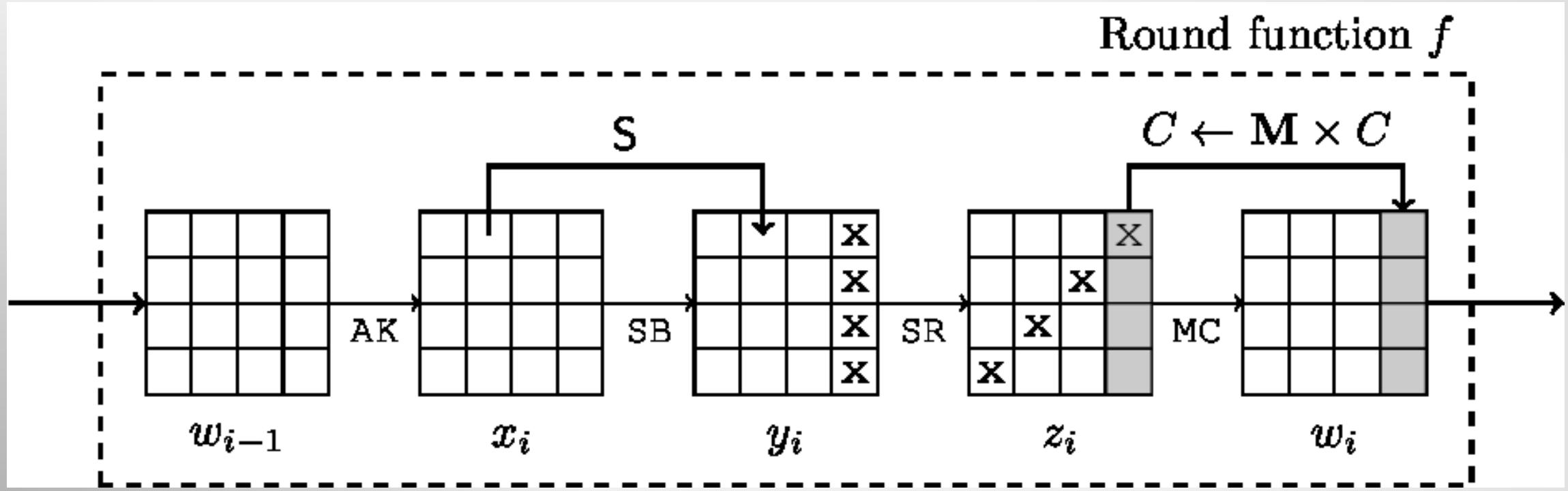
AES Key Schedule

```
__host__ void keyExpansion(u32* key, u32* rk) {
        u32 rk0, rk1, rk2, rk3;
        rk0 = key[0]; rk1 = key[1]; rk2 = key[2]; rk3 = key[3];
        rk[0] = rk0; rk[1] = rk1; rk[2] = rk2; rk[3] = rk3;

        for (u8 roundCount = 0; roundCount < ROUND_COUNT; roundCount++) {
                u32 temp = rk3;
                rk0 = rk0 ^ T4_3[(temp >> 16) & 0xff] ^ T4_2[(temp >> 8) & 0xff] ^
                T4_1[(temp) & 0xff] ^ T4_0[(temp >> 24)] ^ RCON32[roundCount];
                rk1 = rk1 ^ rk0;
                rk2 = rk2 ^ rk1;
                rk3 = rk2 ^ rk3;
                rk[roundCount * 4 + 4] = rk0;
                rk[roundCount * 4 + 5] = rk1;
                rk[roundCount * 4 + 6] = rk2;
                rk[roundCount * 4 + 7] = rk3;
                }
}
```



Lesson 4: Reference C and CUDA Implementation of AES

Advanced Encryption Standard (AES)



Store columns as four 32-bit values

Lesson 4: Reference C and CUDA Implementation of AES

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

S-box of AES

Input $3b$ means the 3rd row and bth column (i.e. $S(3b) = e2$)

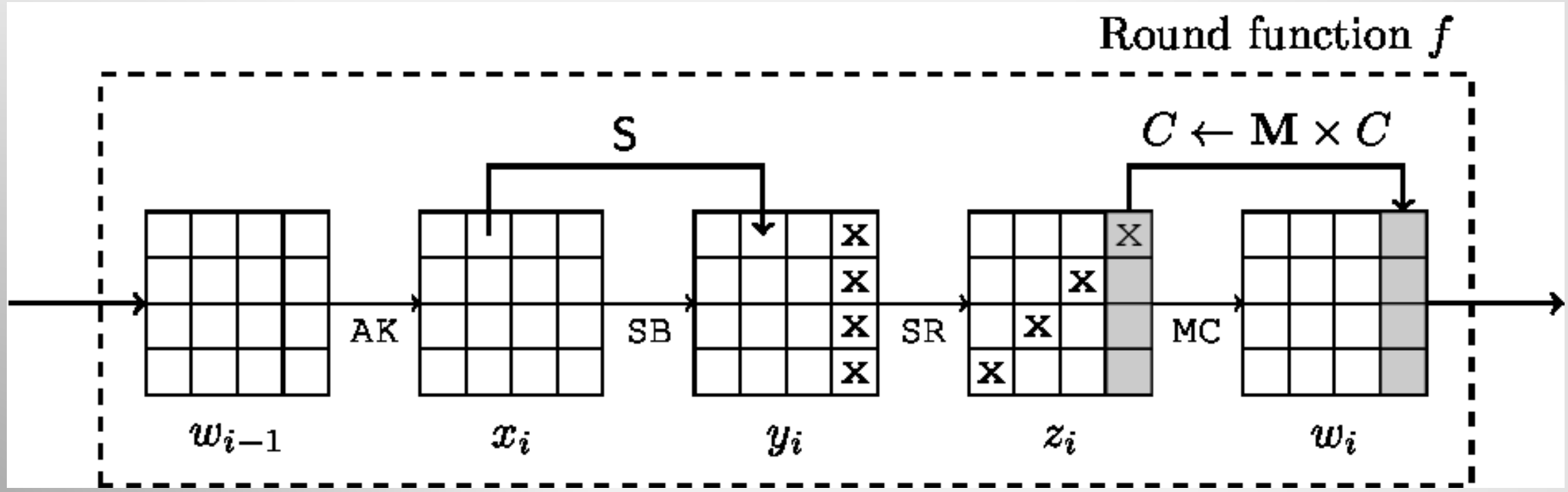
Lesson 4: Reference C and CUDA Implementation of AES

AES S-box as an Array

```
u8 SAES[256] = {0x63,0x7c,0x77,0x7b,0xf2,0x6b,0x6f,0xc5, 0x30, 0x01, 0x67, 0x2b, 0xfe, 0xd7, 0xab,
        0x76,0xca,0x82,0xc9,0x7d,0xfa,0x59,0x47,0xf0, 0xad, 0xd4, 0xa2, 0xaf, 0x9c, 0xa4, 0x72,
        0xc0,0xb7,0xfd,0x93,0x26,0x36,0x3f,0xf7,0xcc, 0x34, 0xa5, 0xe5, 0xf1, 0x71, 0xd8, 0x31,
        0x15,0x04,0xc7,0x23,0xc3,0x18,0x96,0x05,0x9a, 0x07, 0x12, 0x80, 0xe2, 0xeb, 0x27, 0xb2,
        0x75,0x09,0x83,0x2c,0x1a,0x1b,0x6e,0x5a,0xa0, 0x52, 0x3b, 0xd6, 0xb3, 0x29, 0xe3, 0x2f,
        0x84,0x53,0xd1,0x00,0xed,0x20,0xfc,0xb1,0x5b, 0x6a, 0xcb, 0xbe, 0x39, 0x4a, 0x4c, 0x58,
        0xcf,0xd0,0xef,0xaa,0xfb,0x43,0x4d,0x33,0x85, 0x45, 0xf9, 0x02, 0x7f, 0x50, 0x3c, 0x9f,
        0xa8,0x51,0xa3,0x40,0x8f,0x92,0x9d,0x38,0xf5, 0xbc, 0xb6, 0xda, 0x21, 0x10, 0xff, 0xf3,
        0xd2,0xcd,0x0c,0x13,0xec,0x5f,0x97,0x44,0x17, 0xc4, 0xa7, 0x7e, 0x3d, 0x64, 0x5d, 0x19,
        0x73,0x60,0x81,0x4f,0xdc,0x22,0x2a,0x90,0x88, 0x46, 0xee, 0xb8, 0x14, 0xde, 0x5e, 0x0b,
        0xdb,0xe0,0x32,0x3a,0x0a,0x49,0x06,0x24,0x5c, 0xc2, 0xd3, 0xac, 0x62, 0x91, 0x95, 0xe4,
        0x79,0xe7,0xc8,0x37,0x6d,0x8d,0xd5,0x4e,0xa9, 0x6c, 0x56, 0xf4, 0xea, 0x65, 0x7a, 0xae,
        0x08,0xba,0x78,0x25,0x2e,0x1c,0xa6,0xb4,0xc6, 0xe8, 0xdd, 0x74, 0x1f, 0x4b, 0xbd, 0x8b,
        0x8a,0x70,0x3e,0xb5,0x66,0x48,0x03,0xf6,0x0e, 0x61, 0x35, 0x57, 0xb9, 0x86, 0xc1, 0x1d,
        0x9e,0xe1,0xf8,0x98,0x11,0x69,0xd9,0x8e,0x94, 0x9b, 0x1e, 0x87, 0xe9, 0xce, 0x55, 0x28,
        0xdf,0x8c,0xa1,0x89,0x0d,0xbf,0xe6,0x42,0x68, 0x41, 0x99, 0x2d, 0x0f, 0xb0, 0x54, 0xbb, 0x16 };
```

Lesson 4: Reference C and CUDA Implementation of AES

Advanced Encryption Standard (AES)



Lesson 4: Reference C and CUDA Implementation of AES

Mix Columns (MC) *(Omitted in the last round!)*

$$M = \begin{pmatrix} 0x02 & 0x03 & 0x01 & 0x01 \\ 0x01 & 0x02 & 0x02 & 0x01 \\ 0x01 & 0x01 & 0x03 & 0x03 \\ 0x02 & 0x01 & 0x01 & 0x02 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix}$$

Lesson 4: Reference C and CUDA Implementation of AES

AES S-box as an Array

```
u32 multiply2[256]= {
        0x00,0x02,0x04,0x06,0x08,0x0a,0x0c,0x0e,0x10,0x12,0x14,0x16,0x18,0x1a,0x1c,0x1e,
        0x20,0x22,0x24,0x26,0x28,0x2a,0x2c,0x2e,0x30,0x32,0x34,0x36,0x38,0x3a,0x3c,0x3e,
        0x40,0x42,0x44,0x46,0x48,0x4a,0x4c,0x4e,0x50,0x52,0x54,0x56,0x58,0x5a,0x5c,0x5e,
        0x60,0x62,0x64,0x66,0x68,0x6a,0x6c,0x6e,0x70,0x72,0x74,0x76,0x78,0x7a,0x7c,0x7e,
        0x80,0x82,0x84,0x86,0x88,0x8a,0x8c,0x8e,0x90,0x92,0x94,0x96,0x98,0x9a,0x9c,0x9e,
        0xa0,0xa2,0xa4,0xa6,0xa8,0xaa,0xac,0xae,0xb0,0xb2,0xb4,0xb6,0xb8,0xba,0xbc,0xbe,
        0xc0,0xc2,0xc4,0xc6,0xc8,0xca,0xcc,0xce,0xd0,0xd2,0xd4,0xd6,0xd8,0xda,0xdc,0xde,
        0xe0,0xe2,0xe4,0xe6,0xe8,0xea,0xec,0xee,0xf0,0xf2,0xf4,0xf6,0xf8,0xfa,0xfc,0xfe,
        0x1b,0x19,0x1f,0x1d,0x13,0x11,0x17,0x15,0x0b,0x09,0x0f,0x0d,0x03,0x01,0x07,0x05,
        0x3b,0x39,0x3f,0x3d,0x33,0x31,0x37,0x35,0x2b,0x29,0x2f,0x2d,0x23,0x21,0x27,0x25,
        0x5b,0x59,0x5f,0x5d,0x53,0x51,0x57,0x55,0x4b,0x49,0x4f,0x4d,0x43,0x41,0x47,0x45,
        0x7b,0x79,0x7f,0x7d,0x73,0x71,0x77,0x75,0x6b,0x69,0x6f,0x6d,0x63,0x61,0x67,0x65,
        0x9b,0x99,0x9f,0x9d,0x93,0x91,0x97,0x95,0x8b,0x89,0x8f,0x8d,0x83,0x81,0x87,0x85,
        0xbb,0xb9,0xbf,0xbd,0xb3,0xb1,0xb7,0xb5,0xab,0xa9,0xaf,0xad,0xa3,0xa1,0xa7,0xa5,
        0xdb,0xd9,0xdf,0xdd,0xd3,0xd1,0xd7,0xd5,0xcb,0xc9,0xcf,0xcd,0xc3,0xc1,0xc7,0xc5,
        0xfb,0xf9,0xff,0xfd,0xf3,0xf1,0xf7,0xf5,0xeb,0xe9,0xef,0xed,0xe3,0xe1,0xe7,0xe5};
```

Lesson 4: Reference C and CUDA Implementation of AES

AES S-box as an Array

```
u32 multiply3[256]= {
        0x00,0x03,0x06,0x05,0x0c,0x0f,0x0a,0x09,0x18,0x1b,0x1e,0x1d,0x14,0x17,0x12,0x11,
        0x30,0x33,0x36,0x35,0x3c,0x3f,0x3a,0x39,0x28,0x2b,0x2e,0x2d,0x24,0x27,0x22,0x21,
        0x60,0x63,0x66,0x65,0x6c,0x6f,0x6a,0x69,0x78,0x7b,0x7e,0x7d,0x74,0x77,0x72,0x71,
        0x50,0x53,0x56,0x55,0x5c,0x5f,0x5a,0x59,0x48,0x4b,0x4e,0x4d,0x44,0x47,0x42,0x41,
        0xc0,0xc3,0xc6,0xc5,0xcc,0xcf,0xca,0xc9,0xd8,0xdb,0xde,0xdd,0xd4,0xd7,0xd2,0xd1,
        0xf0,0xf3,0xf6,0xf5,0xfc,0xff,0xfa,0xf9,0xe8,0xeb,0xee,0xed,0xe4,0xe7,0xe2,0xe1,
        0xa0,0xa3,0xa6,0xa5,0xac,0xaf,0xaa,0xa9,0xb8,0xbb,0xbe,0xbd,0xb4,0xb7,0xb2,0xb1,
        0x90,0x93,0x96,0x95,0x9c,0x9f,0x9a,0x99,0x88,0x8b,0x8e,0x8d,0x84,0x87,0x82,0x81,
        0x9b,0x98,0x9d,0x9e,0x97,0x94,0x91,0x92,0x83,0x80,0x85,0x86,0x8f,0x8c,0x89,0x8a,
        0xab,0xa8,0xad,0xae,0xa7,0xa4,0xa1,0xa2,0xb3,0xb0,0xb5,0xb6,0xbf,0xbc,0xb9,0xba,
        0xfb,0xf8,0xfd,0xfe,0xf7,0xf4,0xf1,0xf2,0xe3,0xe0,0xe5,0xe6,0xef,0xec,0xe9,0xea,
        0xcb,0xc8,0xcd,0xce,0xc7,0xc4,0xc1,0xc2,0xd3,0xd0,0xd5,0xd6,0xdf,0xdc,0xd9,0xda,
        0x5b,0x58,0x5d,0x5e,0x57,0x54,0x51,0x52,0x43,0x40,0x45,0x46,0x4f,0x4c,0x49,0x4a,
        0x6b,0x68,0x6d,0x6e,0x67,0x64,0x61,0x62,0x73,0x70,0x75,0x76,0x7f,0x7c,0x79,0x7a,
        0x3b,0x38,0x3d,0x3e,0x37,0x34,0x31,0x32,0x23,0x20,0x25,0x26,0x2f,0x2c,0x29,0x2a,
        0x0b,0x08,0x0d,0x0e,0x07,0x04,0x01,0x02,0x13,0x10,0x15,0x16,0x1f,0x1c,0x19,0x1a};
```

Next lecture

CUDA Optimization of AES

- This C reference implementation can be turned into a CUDA code easily
- We are going to see table-based implementation technique for AES
- And we will see a method to remove shared memory bank conflicts

Thanks

This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 101101903. The JU receives support from the Digital Europe Programme and Germany, Bulgaria, Austria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, Greece, Hungary, Ireland, Italy, Lithuania, Latvia, Poland, Portugal, Romania, Slovenia, Spain, Sweden, France, Netherlands, Belgium, Luxembourg, Slovakia, Norway, Türkiye, Republic of North Macedonia, Iceland, Montenegro, Serbia