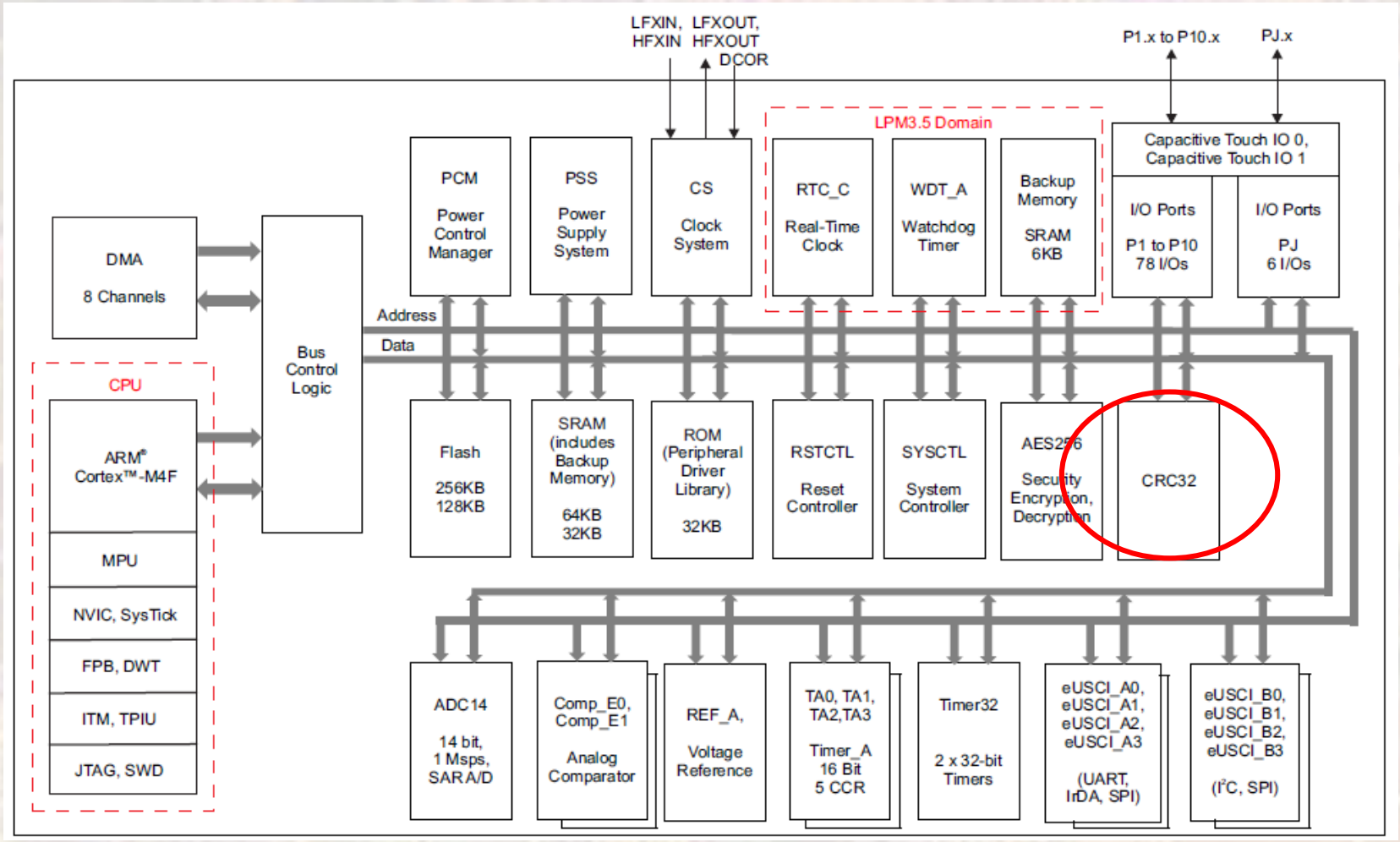
Cyclic Redundancy Check

Last updated 6/17/19

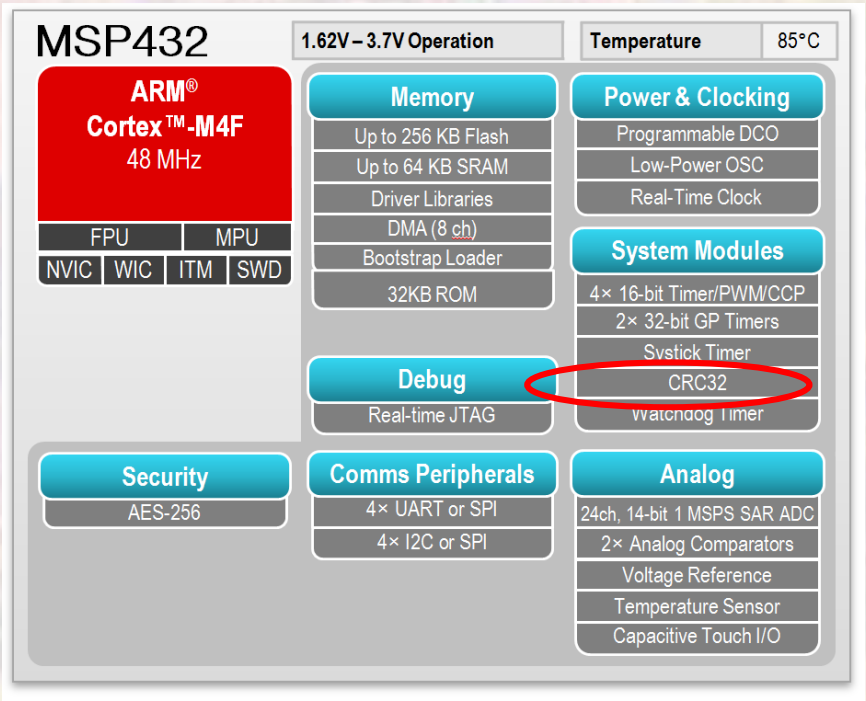
CRC

- MSP432 CRC



CRC

- MSP432 CRC
  - AMBA Compliant
  - 16 and 32 bit variants



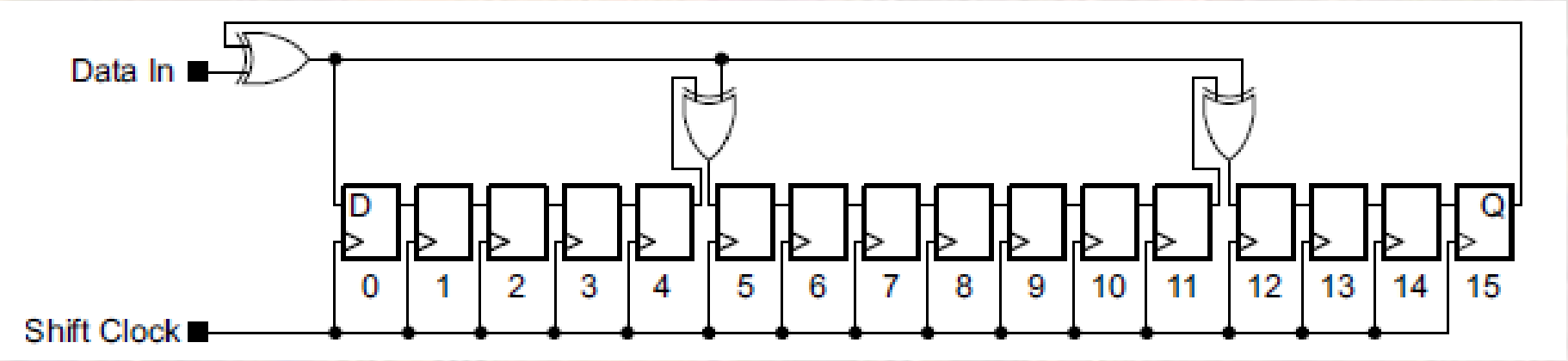
CRC

- MSP432 CRC
  - Creates a signature for validating data
    - Send
      - Data → CRC → send-check-value
      - Send data and send-check-value
    - Receive
      - Separate data and send-check-value
      - Data → CRC -> receive-check-value
    - Match check values
      - Same – no error in transmission
      - Different – error of some type
  - CRC16 - CRC16-CCITT standard
  - CRC32 - CRC32-ISO3309 standard

CRC

- MSP432 CRC
  - CRC16 - CRC16-CCITT standard

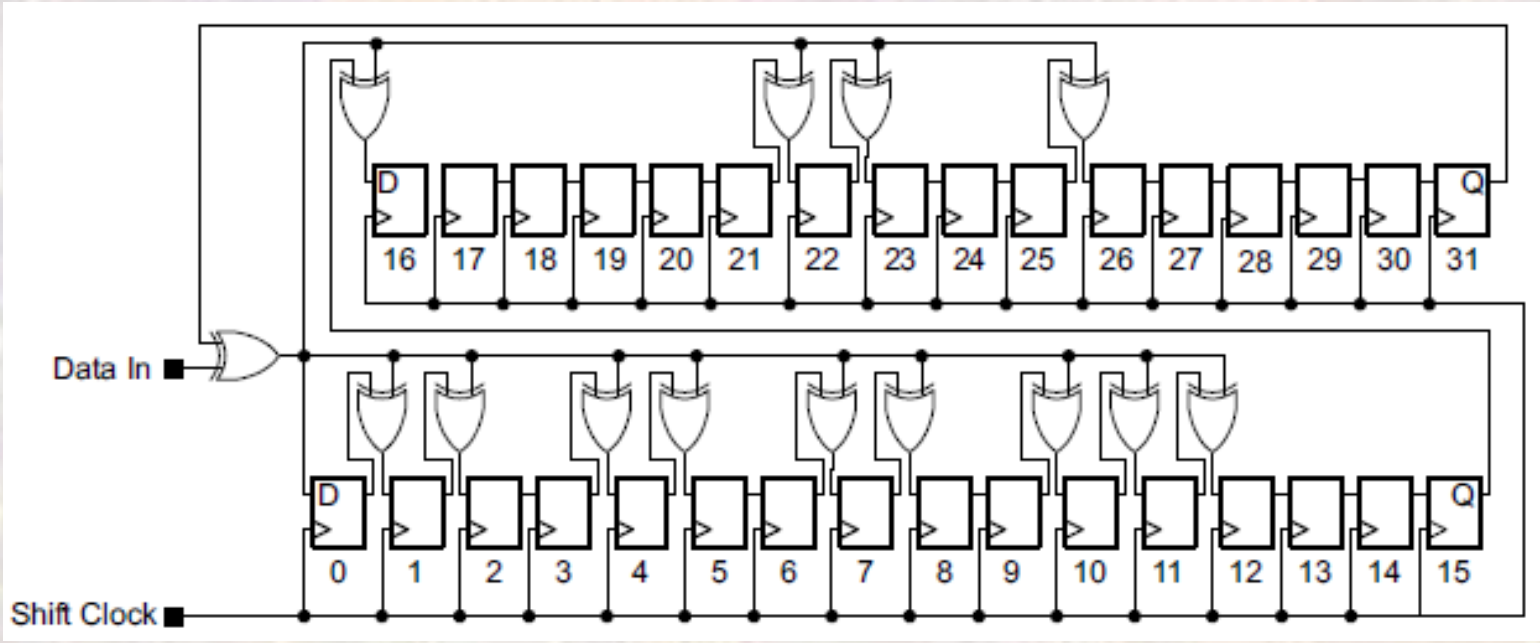
$$f(x) = x_{15} + x_{12} + x_5 + 1$$



CRC

- MSP432 CRC
  - CRC32 - CRC32-ISO3309 standard

$$f(x) = x_{32} + x_{26} + x_{23} + x_{22} + x_{16} + x_{12} + x_{11} + x_{10} + x_8 + x_7 + x_5 + x_4 + x_2 + x + 1$$



CRC

- MSP432 CRC
  - A word about MSBs
  - In the olden days the MSB was labeled bit-0
    - Impacts the CRC calculation
  - A set of reverse registers included

CRC

- MSP432 CRC

Table 13-1. CRC32 Registers

Offset	Acronym	Register Name	Section
0000h	CRC32DI	CRC32 Data Input Low	Section 13.3.0.1
0002h	Reserved	Reserved	
0004h	CRC32DIRB	CRC32 Data In Reverse Low	Section 13.3.0.2
0006h	Reserved	Reserved	
0008h	CRC32INIRES_LO	CRC32 Initialization and Result Low	Section 13.3.0.3
000Ah	CRC32INIRES_HI	CRC32 Initialization and Result High	Section 13.3.0.4
000Ch	CRC32RESR_LO	CRC32 Result Reverse Low	Section 13.3.0.5
000Eh	CRC32RESR_HI	CRC32 Result Reverse High	Section 13.3.0.6
0010h	CRC16DI	CRC16 Data Input Low	Section 13.3.0.7
0012h	Reserved	Reserved	
0014h	CRC16DIRB	CRC16 Data In Reverse Low	Section 13.3.0.8
0016h	Reserved	Reserved	
0018h	CRC16INIRES	CRC16 Init and Result	Section 13.3.0.9
001Ah	Reserved	Reserved	
001Ch	Reserved	Reserved	
001Eh	CRC16RESR	CRC16 Result Reverse	Section 13.3.0.10

CRC

- MSP432 CRC

Figure 13-3. CRC32DI Register

15	14	13	12	11	10	9	8
CRC32DI							
rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0
7	6	5	4	3	2	1	0
CRC32DI							
rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0

Table 13-2. CRC32DI Register Description

Bit	Field	Type	Reset	Description
15-0	CRC32DI	RW	0h	Data input for CRC32 Computation. Data written to the register is included to the present signature in the CRC32INIRES_HI and CRC32INIRES_LO registers according to the CRC32-ISO3309 standard.

CRC

- MSP432 CRC

Figure 13-4. CRC32DIRB Register

15	14	13	12	11	10	9	8
CRC32DIRB							
rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0
7	6	5	4	3	2	1	0
CRC32DIRB							
rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0

Table 13-3. CRC32DIRB Register Description

Bit	Field	Type	Reset	Description
15-0	CRC32DIRB	RW	0h	CRC32 data in bit reversed. Data written to the register is included to the present signature in the CRC32INIRES_HI and CRC32INIRES_LO registers according to the CRC32-ISO3309 standard.

CRC

- MSP432 CRC

Figure 13-5. CRC32INIRES_LO Register

15	14	13	12	11	10	9	8
CRC32INIRES_LO							
rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0
7	6	5	4	3	2	1	0
CRC32INIRES_LO							
rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0

Table 13-4. CRC32INIRES_LO Register Description

Bit	Field	Type	Reset	Description
15-0	CRC32INIRES_LO[1]	RW	0h	CRC32 initialization and result, lower 16 bits. This register holds lower 16 bits of the current CRC32 result (according to the CRC32-ISO3309 standard). Writing to this register initializes the CRC32 calculation with the value written to it.

(1) This register is updated with the final signature value one cycle after the last data input value is written to the CRC32DI or CRC32DIRB registers. Application should wait for this one cycle delay before reading the result.

CRC

- MSP432 CRC

Figure 13-6. CRC32INIRES_HI Register

15	14	13	12	11	10	9	8
CRC32INIRES_HI							
rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0
7	6	5	4	3	2	1	0
CRC32INIRES_HI							
rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0

Table 13-5. CRC32INIRES_HI Register Description

Bit	Field	Type	Reset	Description
15-0	CRC32INIRES_HI (1)	RW	0h	CRC32 initialization and result, upper 16 bits. This register holds the upper 16 bits of the current CRC32 result (according to the CRC32-ISO3309 standard). Writing to this register initializes the CRC32 calculation with the value written to it.

(1) This register is updated with the final signature value one cycle **after** the last data input value is written to the CRC32DI or CRC32DIRB registers. Application should wait for this one cycle delay before reading the result.

CRC

- MSP432 CRC

Figure 13-7. CRC32RESR_LO Register

15	14	13	12	11	10	9	8
CRC32RESR_LO							
rw-1	rw-1	rw-1	rw-1	rw-1	rw-1	rw-1	rw-1
7	6	5	4	3	2	1	0
CRC32RESR_LO							
rw-1	rw-1	rw-1	rw-1	rw-1	rw-1	rw-1	rw-1

Table 13-6. CRC32RESR_LO Register Description

Bit	Field	Type	Reset	Description
15-0	CRC32RESR_LO(1)	RW	FFFFh	CRC32 reverse result, lower 16 bits. This register holds the lower 16 bits of the current CRC32 result (according to the CRC32-ISO3309 standard). The order of bits is reversed to the order of bits in the CRC32INIRES register. The entire 32bit result is reversed and hence both upper and lower result registers must be factored in. For example, CRC32RESR_LO[0] = CRC32INIRES_HI[15], CRC32RESR_LO[10] = CRC32INIRES_HI[5] and so on.

(1) This register is updated with the final signature value one cycle **after** the last data input value is written to the CRC32DI or CRC32DIRB registers. Application should wait for this one cycle delay before reading the result.

CRC

- MSP432 CRC

Figure 13-8. CRC32RESR_HI Register

15	14	13	12	11	10	9	8
CRC32RESR_HI							
rw-1	rw-1	rw-1	rw-1	rw-1	rw-1	rw-1	rw-1
7	6	5	4	3	2	1	0
CRC32RESR_HI							
rw-1	rw-1	rw-1	rw-1	rw-1	rw-1	rw-1	rw-1

Table 13-7. CRC32RESR_HI Register Description

Bit	Field	Type	Reset	Description
15-0	CRC32RESR_HI[(1)]	RW	FFFFh	CRC32 reverse result, upper 16 bits. This register holds the upper 16 bits of the current CRC32 result (according to the CRC32-ISO3309 standard). The order of bits is reversed to the order of bits in the CRC32INIRES register. The entire 32bit result is reversed and hence both upper and lower result registers must be factored in. For example, CRC32RESR_HI[0] = CRC32INIRES_LO[15], CRC32RESR_HI[10] = CRC32INIRES_LO[5] and so on.

(1) This register is updated with the final signature value one cycle **after** the last data input value is written to the CRC32DI or CRC32DIRB registers. Application should wait for this one cycle delay before reading the result.

CRC

- MSP432 CRC

Figure 13-9. CRC16DI Register

15	14	13	12	11	10	9	8
CRC16DI							
rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0
7	6	5	4	3	2	1	0
CRC16DI							
rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0

Table 13-8. CRC16DI Register Description

Bit	Field	Type	Reset	Description
15-0	CRC16DI	RW	0h	CRC16 data in. Data written to the CRC16DI register is included to the present signature in the CRC16INIRES register according to the CRC16-CCITT standard.

CRC

- MSP432 CRC

Figure 13-10. CRC16DIRB Register

15	14	13	12	11	10	9	8
CRC16DIRB							
rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0
7	6	5	4	3	2	1	0
CRC16DIRB							
rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0

Table 13-9. CRC16DIRB Register Description

Bit	Field	Type	Reset	Description
15-0	CRC16DIRB	RW	0h	CRC16 data in reverse byte. Data written to the CRC16DIRB register is included to the present signature in the CRC16INIRES and CRC16RESR registers according to the CRC-CCITT standard. Reading the register returns the register CRC16DI content.

CRC

- MSP432 CRC

Figure 13-11. CRC16INIRES Register

15	14	13	12	11	10	9	8
CRC16INIRES							
rw-1	rw-1	rw-1	rw-1	rw-1	rw-1	rw-1	rw-1
7	6	5	4	3	2	1	0
CRC16INIRES							
rw-1	rw-1	rw-1	rw-1	rw-1	rw-1	rw-1	rw-1

Table 13-10. CRC16INIRES Register Description

Bit	Field	Type	Reset	Description
15-0	CRC16INIRES[1]	RW	FFh	CRC16 initialization and result. This register holds the current CRC16 result (according to the CRC16-CCITT standard). Writing to this register initializes the CRC16 calculation with the value written to it.

(1) This register is updated with the final signature value one cycle **after** the last data input value is written to the CRC32DI or CRC32DIRB registers. Application should wait for this one cycle delay before reading the result.

CRC

- MSP432 CRC

Figure 13-12. CRC16RESR Register

15	14	13	12	11	10	9	8
CRC16RESR							
rw-1	rw-1	rw-1	rw-1	rw-1	rw-1	rw-1	rw-1
7	6	5	4	3	2	1	0
CRC16RESR							
rw-1	rw-1	rw-1	rw-1	rw-1	rw-1	rw-1	rw-1

Table 13-11. CRC16RESR Register Description

Bit	Field	Type	Reset	Description
15-0	CRC16RESR[(1)]	RW	FFh	CRC16 reverse result. This register holds the current CRC16 result (according to the CRC16-CCITT standard). The order of bits is reversed to the order of bits in the CRC16INIRES register. For example, CRC16RESR[0] = CRC16INIRES[15], CRC32RESR[10] = CRC16INIRES[5] and so on.

(1) This register is updated with the final signature value one cycle **after** the last data input value is written to the CRC32DI or CRC32DIRB registers. Application should wait for this one cycle delay before reading the result.

CRC

- MSP432 CRC

```
/*
 * crc.c
 *
 *  Created on: Oct 22, 2017
 *      Author: johnsontimoj
 */
////////////////////////////
// crc test file
//
// Simulates data transmission
// with and w/o errors
//
////////////////////////////

#include <stdio.h>
#include <stdint.h>
#include "msp432.h"
#include "msoe_lib_all.h"
```

```
int main(void){
    //
    // Use CRC16 with
    // Key = 0x1357

    //
    // Send Side
    //
    // Load seed
    CRC32->INIRES16 = 0x1357;
    // Load Data in
    CRC32->DI16 = 0x2468;
    CRC32->DI16 = 0x8642;
    CRC32->DI16 = 0x1357;
    CRC32->DI16 = 0x7531;
    // wait one clock
    ;
    // Print send-check-value
    printf("\nsend-check-value is: 0x%x\n",CRC32->INIRES16);
    //
    // The sender would send the data followed by the send-check-value
    // 0x2468864213577531xxxx, in this case xxxx is 2f46
    //
```

CRC

- MSP432 CRC

```
    //
    // Receive Side
    //
    // The receiver receives
    // 0x2468642135775312f46   NO ERRORS
    //
    // Load seed
    CRC32->INIRES16 = 0x1357;
    // Load Data in
    CRC32->DI16 = 0x2468;
    CRC32->DI16 = 0x8642;
    CRC32->DI16 = 0x1357;
    CRC32->DI16 = 0x7531;
    // wait one clock
    ;
    printf("\nNo error version\n");
    printf("receive-check-value is: 0x%x\n",CRC32->INIRES16);
    //
    // values should match
    //
    // Receive Side
    //
    // The receiver receives
    // 0x24688642135775302f46   Error on Nibble 4 (is 0, should be 1)
    //
    // Load seed
    CRC32->INIRES16 = 0x1357;
    // Load Data in
    CRC32->DI16 = 0x2468;
    CRC32->DI16 = 0x8642;
    CRC32->DI16 = 0x1357;
    CRC32->DI16 = 0x7530;
    // wait one clock
    ;
    printf("\nVersion with error\n");
    printf("receive-check-value is: 0x%x\n",CRC32->INIRES16);
    //
    // Values should not match --> error

    return 0;

} // end main
```

```
Class_Project:CIO
[CORTEX_M4_0]
send-check-value is: 0x2f46

No error version
receive-check-value is: 0x2f46

Version with error
receive-check-value is: 0x34de
```

Match

No Match