**Activity 4.O – Study of shift and rotate instructions using encoding and decoding**

**Goal of Lab: Implement Hamming code at the architecture level. Extends in class activity 5**

**Operators:**

**Sample Lab:**

**DECODING**

**TUTOR 1.32> MM 3000;DI**

**003000 203809C4 MOVE.L $000009C4,D0 ?**

**003004 4241 CLR.W D1 ?**

**003006 243C00000002 MOVE.L #2,D2 ?**

**00300C 263C00000004 MOVE.L #4,D3 ?**

**003012 283C00000008 MOVE.L #8,D4 ?**

**003018 2A3C00000010 MOVE.L #16,D5 ?**

**00301E 2C3C00000020 MOVE.L #32,D6 ?**

**003024 2E3C00000040 MOVE.L #64,D7 ?**

**00302A CE40 AND.W D0,D7 ?**

**00302C CC40 AND.W D0,D6 ?**

**00302E CA40 AND.W D0,D5 ?**

**003030 C840 AND.W D0,D4 ?**

**003032 C640 AND.W D0,D3 ?**

**003034 C440 AND.W D0,D2 ?**

**003036 ED4F LSL.W #6,D7 ?**

**003038 BF41 EOR.W D7,D1 ?**

**00303A EA4E LSR.W #5,D6 ?**

**00303C BD41 EOR.W D6,D1 ?**

**00303E E84D LSR.W #4,D5 ?**

**003040 BB41 EOR.W D5,D1 ?**

**003042 E64C LSR.W #3,D4 ?**

**003044 B941 EOR.W D4,D1 ?**

**003046 E349 LSL.W #1,D1 ?**

**003048 BF41 EOR.W D7,D1 ?**

**00304A BD41 EOR.W D6,D1 ?**

**00304C E44B LSR.W #2,D3 ?**

**00304E B741 EOR.W D3,D1 ?**

**003050 E24A LSR.W #1,D2 ?**

**003052 B541 EOR.W D2,D1 ?**

**003054 E349 LSL #1,D1**

**003056 BF41 EOR D7,D1**

**003058 BB41 EOR D5,D1**

**00305A B741 EOR D3,D1**

**00305C 243C00000001 MOVE.L #1,D2**

**003062 C440 AND D0,D2**

**003064 B541 EOR D2,D1**

**003066 2001 MOVE.L D1,D0**

**003068 00000000 OR.B #0,D0 ?.**

**TUTOR 1.32> BR 3068**

**BREAKPOINTS**

**003068 003068**

**002050 002050**

**TUTOR 1.32>**

**TUTOR 1.32> MM 2000;DI**

**002000 43F82400 LEA.L $00002400,A1 ?**

**002004 4241 CLR.W D1 ?**

**002006 1211 MOVE.B (A1),D1 ?**

**002008 243C00000001 MOVE.L #1,D2 ?**

**00200E 263C00000002 MOVE.L #2,D3**

**002014 283C00000004 MOVE.L #4,D4**

**00201A 2A3C00000008 MOVE.L #8,D5**

**002020 ANDW**

**002020 AND.W D1.D2**

**002020 C641 AND.W D1,D3**

**002022 C841 AND.W D1,D4**

**002024 CA41 AND.W D1,D5**

**002026 E249 LSR.W #1,D1**

**002028 E24B LSR.W #1,D3**

**00202A LSR.W #2.D4**

**00202A E44C LSR.W #2,D4**

**00202C E64D LSR.W #3,D5**

**00202E 2004 MOVE.L D4,D0**

**002030 BB40 EOR.W D5,D0**

**002032 2800 MOVE.L D0,D4**

**002034 B744 EOR.W D3,D4**

**002036 E349 LSL.W #1,D1**

**002038 B941 EOR.W D4,D1**

**00203A E349 LSL.W #1,D1**

**00203C B541 EOR.W D2,D1**

**00203E E349 LSL.W #1,D1**

**002040 B540 EOR.W D2,D0**

**002042 B141 EOR.W D0,D1**

**002044 E349 LSL.W #1,D1**

**002046 BB43 EOR.W D5,D3**

**002048 B543 EOR.W D2,D3**

**00204A B741 EOR.W D3,D1**

**00204C 21C109C4 MOVE.L D1,2500**

**002050 9FFF DC.W $9FFF ?.**

**TUTOR 1.32> BR 2050**

**TUTOR 1.32> .PC 2000**

**TUTOR 1.32> MS 2400 0B**

**TUTOR 1.32> GO**

**PHYSICAL ADDRESS=00002000**

**AT BREAKPOINT**

**PC=00002050 SR=2700=.S7..... US=FFFF0001 SS=000002CC**

**D0=00000000 D1=00000055 D2=00000001 D3=00000001**

**D4=00000000 D5=00000001 D6=00000000 D7=00000003**

**A0=FDF7FFFF A1=00002400 A2=FFDFFFFF A3=01830003**

**A4=FFFFFFFB A5=2B7B7EFF A6=FFFFFFFE A7=000002CC**

**--------------------002050 9FFF DC.W $9FFF**

**TUTOR 1.32> .PC 3000**

**TUTOR 1.32> GO**

**PHYSICAL ADDRESS=00003000**

**AT BREAKPOINT**

**PC=00003068 SR=2700=.S7..... US=FFFF0001 SS=000002CC**

**D0=00007007 D1=00007007 D2=00000001 D3=00000001**

**D4=00000000 D5=00000001 D6=00000000 D7=00001000**

**A0=FDF7FFFF A1=00002400 A2=FFDFFFFF A3=01830003**

**A4=FFFFFFFB A5=2B7B7EFF A6=FFFFFFFE A7=000002CC**

**--------------------003068 00000000 OR.B #0,D0**

**TUTOR 1.32> MS 9C4 0000 0045**

**TUTOR 1.32> .PC 3000**

**TUTOR 1.32> GO**

**PHYSICAL ADDRESS=00003000**

**AT BREAKPOINT**

**PC=00003068 SR=2700=.S7..... US=FFFF0001 SS=000002CC**

**D0=00007002 D1=00007002 D2=00000001 D3=00000001**

**D4=00000000 D5=00000000 D6=00000000 D7=00001000**

**A0=FDF7FFFF A1=00002400 A2=FFDFFFFF A3=01830003**

**A4=FFFFFFFB A5=2B7B7EFF A6=FFFFFFFE A7=000002CC**

**--------------------003068 00000000 OR.B #0,D0**

**TUTOR 1.32> MS 9C4 0000 003E**

**TUTOR 1.32> .PC 3000**

**TUTOR 1.32> GO**

**PHYSICAL ADDRESS=00003000**

**AT BREAKPOINT**

**PC=00003068 SR=2700=.S7..... US=FFFF0001 SS=000002CC**

**D0=00000006 D1=00000006 D2=00000000 D3=00000001**

**D4=00000001 D5=00000001 D6=00000001 D7=00000000**

**A0=FDF7FFFF A1=00002400 A2=FFDFFFFF A3=01830003**

**A4=FFFFFFFB A5=2B7B7EFF A6=FFFFFFFE A7=000002CC**

**--------------------003068 00000000 OR.B #0,D0**

1. Lab 4.O – Hamming Code

DESCRIPTION

The Hamming code is a special technique for encoding and decoding information to enable error detection and correction. Richard Hamming published his work in 1951. The high cost of hardware in the past made this technique costly and, therefore, not practical. A CRAY computer design was released to the marketplace which uses single error correcting techniques as described in the Hamming paper.

ENCODING

A significant feature of the Hamming code is that all components of the algorithm are mathematically sound and practical. The encoding procedure is to compute “even parity” bits based on the information to be transmitted and send those parity bits along with the transmitted information. If the information contains 4 bits (abcd), then three parity bits (rst) are needed. These parity bits are called check bits. The encoded message is then structured as:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | position | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
|  | bit | a | b | c | r | d | s | t |

where:

r is set to create even parity for bits 7, 6, 5 and 4;

s is set to create even parity for bits 7, 6, 3 and 2;

t is set to create even parity for bits 7, 5, 3 and 1.

You are to encode the contents of a byte in memory where the information bits a, b, c and d are located as shown in the following:

xxxxabcd

For example, suppose that you have the following memory configuration, your task would be to compose the encoded bytes as shown in Table 4.1.

Table 4.1 Examples of messages and their encoded messages using Hamming code.

|  |  |  |
| --- | --- | --- |
|  | **Memory** | **Encoded Message** |
| 2274 | xxxx1011 | 01010101 |
| 2275 | xxxx1001 | 01001100 |
| 2276 | xxxx0011 | 00011110 |

ALGORITHM

Goal: Convert a given message into an encoded message using Hamming encoding.

Picture the encoded message as shown in Table 4.2, and verify the accuracy of the results using Table 4.1.

R, S, and T are computed using the following equations:

R = A ⊕ B ⊕ C

S = A ⊕ B ⊕ D

T = A ⊕ C ⊕ D

Table 4.2 Dichotomy of an original message and its encoded equivalence.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Bit Position | 4 | 3 | 2 | 1 | 🡪  🡪 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| Bits | A | B | C | D |  | A | B | C | R | D | S | T |
| Original Message 🡪 Encoded Message | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |

1. Isolate bits A, B, C, D, R, S, T into its own data registers, by masking all other bits other than the bit of interest:

Ex: To get bit A, do:

D0 🡪 00001011

D1 🡪 00001000 ∩

D1 🡪 00001000

To get bit B, do:

D0 🡪 00001011

D2 🡪 00000100 ∩

D2 🡪 00000000

Similar case for C and D, except that the value in register D1 changes.

1. Shift right *x* number of bits.

Ex: D1 🡪 00001000

To get the isolated bit A, we want to get 1 to the right-most end (Least Significant Bit), so shift right by 4 bits.

D1 🡪 00000001

Do the same for B and C with the appropriate number of bits. D is already the LSB, so no shifting is required for D. Make sure to store A, B, C, and D in different data registers, such as D1, D2, D3, and D4.

1. Find R, S, T using these equations, and store them in different data registers.

R = A ⊕ B ⊕ C

S = A ⊕ B ⊕ D

T = A ⊕ C ⊕ D

1. You now have all the bits isolated out in their own data registers. The next step is to append them into one data register by shifting left by one bit, and performing an EOR operation.

Ex: A 🡪 A0 🡪 A0 ⊕ B 🡺 AB

AB 🡪 AB0 🡪 AB0 ⊕ C 🡺 ABC … Continue until you end with ABCRDST

Obviously, in EASy68K you are going to perform these shifting and EOR-ing using the data registers, so make sure to comment your code – it helps in debugging.

*SOURCE CODE*

ORG $1000

START: ; first instruction of program

MOVE.L #0,D2

MOVE.L #0,D3

MOVE.L #0,D4

MOVE.L #0,D5

MOVE.L #0,D6

MOVE.L #0,D7

MOVE.B #11,$00002400

LEA.L $00002400,A0

MOVE.B (A0),D1

MOVE.L #1,D2

MOVE.L #2,D3

MOVE.L #4,D4

MOVE.L #8,D5

AND.W D1,D2 \*D2 = D

AND.W D1,D3 \*D3 = C

AND.W D1,D4 \*D4 = B

AND.W D1,D5 \*D5 = A

LSR.W #1,D3

LSR.W #2,D4

LSR.W #3,D5

MOVE.B D5,D0 \*D0 = A

EOR.W D4,D5

EOR.W D3,D5 \*D5 = R

LSR.W #1,D1

LSL.W #1,D1

EOR.W D5,D1 \*D1 = ABCR

LSL.W #1,D1 \*D1 = ABCR0

EOR.W D2,D1 \*D1 = ABCRD

LSL.W #1,D1 \*D1 = ABCRD0

EOR.W D0,D4

EOR.W D2,D4 \*D4 = S

EOR.W D0,D3

EOR.W D2,D3 \*D3 = T

EOR.W D4,D1 \*D1 = ABCRDS

LSL.W #1,D1 \*D1 = ABCRDS0

EOR.W D3,D1 \*D1 = ABCRDST

\* Put program code here

SIMHALT ; halt simulator

\* Put variables and constants here

END START ; last line of source

*OUTPUT IN TRACE MODE*

D0=00000000 D4=00000000 A0=00000000 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=00000000 D5=00000000 A1=00000000 A5=00000000 SR=0010000000000000

D2=00000000 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000000 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=00001000 Code=7400 Line= 10 MOVE.L #0,D2

//Initializes the register D2 to 0

D0=00000000 D4=00000000 A0=00000000 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=00000000 D5=00000000 A1=00000000 A5=00000000 SR=0010000000000100

D2=00000000 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000000 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=00001002 Code=7600 Line= 11 MOVE.L #0,D3

//Initializes the register D3 to 0

D0=00000000 D4=00000000 A0=00000000 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=00000000 D5=00000000 A1=00000000 A5=00000000 SR=0010000000000100

D2=00000000 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000000 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=00001004 Code=7800 Line= 12 MOVE.L #0,D4

//Initializes the register D4 to 0

D0=00000000 D4=00000000 A0=00000000 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=00000000 D5=00000000 A1=00000000 A5=00000000 SR=0010000000000100

D2=00000000 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000000 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=00001006 Code=7A00 Line= 13 MOVE.L #0,D5

//Initializes the register D5 to 0

D0=00000000 D4=00000000 A0=00000000 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=00000000 D5=00000000 A1=00000000 A5=00000000 SR=0010000000000100

D2=00000000 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000000 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=00001008 Code=7C00 Line= 14 MOVE.L #0,D6

//Initializes the register D6 to 0

D0=00000000 D4=00000000 A0=00000000 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=00000000 D5=00000000 A1=00000000 A5=00000000 SR=0010000000000100

D2=00000000 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000000 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=0000100A Code=7E00 Line= 15 MOVE.L #0,D7

//Initializes the register D7 to 0

D0=00000000 D4=00000000 A0=00000000 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=00000000 D5=00000000 A1=00000000 A5=00000000 SR=0010000000000100

D2=00000000 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000000 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=0000100C Code=11FC 000B 2400 Line= 16 MOVE.B #11,$00002400

//Initializes the address $2400 to decimal number 11 or hex value 0B

D0=00000000 D4=00000000 A0=00000000 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=00000000 D5=00000000 A1=00000000 A5=00000000 SR=0010000000000000

D2=00000000 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000000 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=00001012 Code=41F8 2400 Line= 17 LEA.L $00002400,A0

//Puts the value in $2400 into A0

D0=00000000 D4=00000000 A0=00002400 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=00000000 D5=00000000 A1=00000000 A5=00000000 SR=0010000000000000

D2=00000000 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000000 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=00001016 Code=1210 Line= 18 MOVE.B (A0),D1

//Puts the value in A0 into D1

D0=00000000 D4=00000000 A0=00002400 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=0000000B D5=00000000 A1=00000000 A5=00000000 SR=0010000000000000

D2=00000000 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000000 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=00001018 Code=7401 Line= 19 MOVE.L #1,D2

//Initializes the register D2 to decimal value 1

D0=00000000 D4=00000000 A0=00002400 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=0000000B D5=00000000 A1=00000000 A5=00000000 SR=0010000000000000

D2=00000001 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000000 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=0000101A Code=7602 Line= 20 MOVE.L #2,D3

//Initializes the register D3 to decimal value 2

D0=00000000 D4=00000000 A0=00002400 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=0000000B D5=00000000 A1=00000000 A5=00000000 SR=0010000000000000

D2=00000001 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000002 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=0000101C Code=7804 Line= 21 MOVE.L #4,D4

//Initializes the register D4 to decimal value 4

D0=00000000 D4=00000004 A0=00002400 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=0000000B D5=00000000 A1=00000000 A5=00000000 SR=0010000000000000

D2=00000001 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000002 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=0000101E Code=7A08 Line= 22 MOVE.L #8,D5

//Initializes the register D5 to decimal value 8

D0=00000000 D4=00000004 A0=00002400 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=0000000B D5=00000008 A1=00000000 A5=00000000 SR=0010000000000000

D2=00000001 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000002 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=00001020 Code=C441 Line= 23 AND.W D1,D2

//Performs the AND function to registers D1 and D2 and puts the result in D2

D0=00000000 D4=00000004 A0=00002400 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=0000000B D5=00000008 A1=00000000 A5=00000000 SR=0010000000000000

D2=00000001 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000002 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=00001022 Code=C641 Line= 24 AND.W D1,D3

//Performs the AND function to registers D1 and D3 and puts the result in D3

D0=00000000 D4=00000004 A0=00002400 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=0000000B D5=00000008 A1=00000000 A5=00000000 SR=0010000000000000

D2=00000001 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000002 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=00001024 Code=C841 Line= 25 AND.W D1,D4

//Performs the AND function to registers D1 and D4 and puts the result in D4

D0=00000000 D4=00000000 A0=00002400 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=0000000B D5=00000008 A1=00000000 A5=00000000 SR=0010000000000100

D2=00000001 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000002 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=00001026 Code=CA41 Line= 26 AND.W D1,D5

//Performs the AND function to registers D1 and D5 and puts the result in D5

D0=00000000 D4=00000000 A0=00002400 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=0000000B D5=00000008 A1=00000000 A5=00000000 SR=0010000000000000

D2=00000001 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000002 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=00001028 Code=E24B Line= 27 LSR.W #1,D3

//Shifts the contents in the register D3 by 1 bit-position to the right

D0=00000000 D4=00000000 A0=00002400 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=0000000B D5=00000008 A1=00000000 A5=00000000 SR=0010000000000000

D2=00000001 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000001 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=0000102A Code=E44C Line= 28 LSR.W #2,D4

//Shifts the contents in the register D4 by 2 bit-positions to the right

D0=00000000 D4=00000000 A0=00002400 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=0000000B D5=00000008 A1=00000000 A5=00000000 SR=0010000000000100

D2=00000001 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000001 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=0000102C Code=E64D Line= 29 LSR.W #3,D5

//Shifts the contents in the register D5 by 3 bit-positions to the right

D0=00000000 D4=00000000 A0=00002400 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=0000000B D5=00000001 A1=00000000 A5=00000000 SR=0010000000000000

D2=00000001 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000001 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=0000102E Code=1005 Line= 31 MOVE.B D5,D0

//Moves the contents in the register D5 into D0

D0=00000001 D4=00000000 A0=00002400 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=0000000B D5=00000001 A1=00000000 A5=00000000 SR=0010000000000000

D2=00000001 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000001 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=00001030 Code=B945 Line= 33 EOR.W D4,D5

//Performs the EOR function to the register D4 and D5 and stores the result in D5

D0=00000001 D4=00000000 A0=00002400 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=0000000B D5=00000001 A1=00000000 A5=00000000 SR=0010000000000000

D2=00000001 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000001 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=00001032 Code=B745 Line= 34 EOR.W D3,D5

//Performs the EOR function to the register D3 and D5 and stores the result in D5

D0=00000001 D4=00000000 A0=00002400 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=0000000B D5=00000000 A1=00000000 A5=00000000 SR=0010000000000100

D2=00000001 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000001 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=00001034 Code=E249 Line= 35 LSR.W #1,D1

//Shifts the contents in the register D1 by 1 bit-position to the right

D0=00000001 D4=00000000 A0=00002400 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=00000005 D5=00000000 A1=00000000 A5=00000000 SR=0010000000010001

D2=00000001 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000001 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=00001036 Code=E349 Line= 36 LSL.W #1,D1

//Shifts the contents in the register D1 by 1 bit-position to the left

D0=00000001 D4=00000000 A0=00002400 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=0000000A D5=00000000 A1=00000000 A5=00000000 SR=0010000000000000

D2=00000001 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000001 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=00001038 Code=BB41 Line= 37 EOR.W D5,D1

//Performs the EOR function to the register D1 and D5 and stores the result in D5

D0=00000001 D4=00000000 A0=00002400 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=0000000A D5=00000000 A1=00000000 A5=00000000 SR=0010000000000000

D2=00000001 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000001 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=0000103A Code=E349 Line= 38 LSL.W #1,D1

//Shifts the contents in the register D1 by 1 bit-position to the left

D0=00000001 D4=00000000 A0=00002400 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=00000014 D5=00000000 A1=00000000 A5=00000000 SR=0010000000000000

D2=00000001 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000001 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=0000103C Code=B541 Line= 39 EOR.W D2,D1

//Performs the EOR function to the register D1 and D2 and stores the result in D1

D0=00000001 D4=00000000 A0=00002400 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=00000015 D5=00000000 A1=00000000 A5=00000000 SR=0010000000000000

D2=00000001 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000001 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=0000103E Code=E349 Line= 40 LSL.W #1,D1

//Shifts the contents in the register D1 by 1 bit-position to the left

D0=00000001 D4=00000000 A0=00002400 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=0000002A D5=00000000 A1=00000000 A5=00000000 SR=0010000000000000

D2=00000001 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000001 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=00001040 Code=B144 Line= 42 EOR.W D0,D4

//Performs the EOR function to the register D0 and D4 and stores the result in D4

D0=00000001 D4=00000001 A0=00002400 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=0000002A D5=00000000 A1=00000000 A5=00000000 SR=0010000000000000

D2=00000001 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000001 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=00001042 Code=B544 Line= 43 EOR.W D2,D4

//Performs the EOR function to the register D2 and D4 and stores the result in D4

D0=00000001 D4=00000000 A0=00002400 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=0000002A D5=00000000 A1=00000000 A5=00000000 SR=0010000000000100

D2=00000001 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000001 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=00001044 Code=B143 Line= 45 EOR.W D0,D3

//Performs the EOR function to the register D0 and D3 and stores the result in D3

D0=00000001 D4=00000000 A0=00002400 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=0000002A D5=00000000 A1=00000000 A5=00000000 SR=0010000000000100

D2=00000001 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000000 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=00001046 Code=B543 Line= 46 EOR.W D2,D3

//Performs the EOR function to the register D2 and D3 and stores the result in D3

D0=00000001 D4=00000000 A0=00002400 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=0000002A D5=00000000 A1=00000000 A5=00000000 SR=0010000000000000

D2=00000001 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000001 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=00001048 Code=B941 Line= 48 EOR.W D4,D1

//Performs the EOR function to the register D1 and D4 and stores the result in D1

D0=00000001 D4=00000000 A0=00002400 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=0000002A D5=00000000 A1=00000000 A5=00000000 SR=0010000000000000

D2=00000001 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000001 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=0000104A Code=E349 Line= 49 LSL.W #1,D1

//Shifts the contents in the register D1 by 1 bit-position to the left

D0=00000001 D4=00000000 A0=00002400 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=00000054 D5=00000000 A1=00000000 A5=00000000 SR=0010000000000000

D2=00000001 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000001 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=0000104C Code=B741 Line= 50 EOR.W D3,D1

//Performs the EOR function to the register D1 and D3 and stores the result in D1

D0=00000001 D4=00000000 A0=00002400 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=00000055 D5=00000000 A1=00000000 A5=00000000 SR=0010000000000000

D2=00000001 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000001 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=0000104E Code=FFFF FFFF Line= 75 SIMHALT ; halt simulator

*TEST RUN #1*

//Initializes the registers D1 through D7 to 0

D0=00000000 D4=00000000 A0=00000000 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=00000000 D5=00000000 A1=00000000 A5=00000000 SR=0010000000000000

D2=00000000 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000000 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=00001000 Code=7400 Line= 10 MOVE.L #0,D2

D0=00000000 D4=00000000 A0=00000000 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=00000000 D5=00000000 A1=00000000 A5=00000000 SR=0010000000000100

D2=00000000 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000000 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=00001002 Code=7600 Line= 11 MOVE.L #0,D3

D0=00000000 D4=00000000 A0=00000000 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=00000000 D5=00000000 A1=00000000 A5=00000000 SR=0010000000000100

D2=00000000 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000000 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=00001004 Code=7800 Line= 12 MOVE.L #0,D4

D0=00000000 D4=00000000 A0=00000000 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=00000000 D5=00000000 A1=00000000 A5=00000000 SR=0010000000000100

D2=00000000 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000000 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=00001006 Code=7A00 Line= 13 MOVE.L #0,D5

D0=00000000 D4=00000000 A0=00000000 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=00000000 D5=00000000 A1=00000000 A5=00000000 SR=0010000000000100

D2=00000000 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000000 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=00001008 Code=7C00 Line= 14 MOVE.L #0,D6

D0=00000000 D4=00000000 A0=00000000 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=00000000 D5=00000000 A1=00000000 A5=00000000 SR=0010000000000100

D2=00000000 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000000 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=0000100A Code=7E00 Line= 15 MOVE.L #0,D7

D0=00000000 D4=00000000 A0=00000000 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=00000000 D5=00000000 A1=00000000 A5=00000000 SR=0010000000000100

D2=00000000 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000000 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=0000100C Code=11FC 0009 2400 Line= 16 MOVE.B #9,$00002400

//Initializes the address $00002400 to decimal value 9

00002400: 09 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------

.

.

.

//Program in execution

.

.

.

D0=00000001 D4=00000000 A0=00002400 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=0000004C D5=00000001 A1=00000000 A5=00000000 SR=0010000000000000

D2=00000001 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000000 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=0000104E Code=FFFF FFFF Line= 75 SIMHALT ; halt simulator

//This results in the value of 0000004C in the register D1 which is the encoded result.

*Test Run #2:*

//Initializes all the data registers that is D1 to D7 to decimal value 0

D0=00000000 D4=00000000 A0=00000000 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=00000000 D5=00000000 A1=00000000 A5=00000000 SR=0010000000000000

D2=00000000 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000000 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=00001000 Code=7400 Line= 10 MOVE.L #0,D2

D0=00000000 D4=00000000 A0=00000000 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=00000000 D5=00000000 A1=00000000 A5=00000000 SR=0010000000000100

D2=00000000 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000000 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=00001002 Code=7600 Line= 11 MOVE.L #0,D3

D0=00000000 D4=00000000 A0=00000000 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=00000000 D5=00000000 A1=00000000 A5=00000000 SR=0010000000000100

D2=00000000 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000000 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=00001004 Code=7800 Line= 12 MOVE.L #0,D4

D0=00000000 D4=00000000 A0=00000000 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=00000000 D5=00000000 A1=00000000 A5=00000000 SR=0010000000000100

D2=00000000 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000000 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=00001006 Code=7A00 Line= 13 MOVE.L #0,D5

D0=00000000 D4=00000000 A0=00000000 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=00000000 D5=00000000 A1=00000000 A5=00000000 SR=0010000000000100

D2=00000000 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000000 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=00001008 Code=7C00 Line= 14 MOVE.L #0,D6

D0=00000000 D4=00000000 A0=00000000 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=00000000 D5=00000000 A1=00000000 A5=00000000 SR=0010000000000100

D2=00000000 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000000 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=0000100A Code=7E00 Line= 15 MOVE.L #0,D7

D0=00000000 D4=00000000 A0=00000000 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=00000000 D5=00000000 A1=00000000 A5=00000000 SR=0010000000000100

D2=00000000 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000000 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=0000100C Code=11FC 0003 2400 Line= 16 MOVE.B #3,$00002400

//Initializes the address $2400 to decimal value 3

00002400: 03 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ----------------

.

.

.

//Program in Trace Mode

.

.

.

D0=00000000 D4=00000001 A0=00002400 A4=00000000 T\_S\_\_INT\_\_\_XNZVC

D1=0000001E D5=00000001 A1=00000000 A5=00000000 SR=0010000000000000

D2=00000001 D6=00000000 A2=00000000 A6=00000000 US=00FF0000

D3=00000000 D7=00000000 A3=00000000 A7=01000000 SS=01000000

PC=0000104E Code=FFFF FFFF Line= 75 SIMHALT ; halt simulator

//This results in the value of 0000004C in the register D1 which is the encoded result.