Computer Architecture

Assignment #1: ARM Instructions Analysis

Binary number

Binary number represents any number with 0 and 1

How to convert Decimal to Binary?

7392 = 7 × 103 + 3 × 102 + 9 × 101 +2 × 100

How to convert binary to Decimal?

(11010)2 =1×24+1×23+0×22+1×21+0×20 = (26)10

210 = 1Kilo

220 = 1Mega

230 = 1Giga

Convert to binary

- Ex 1-1) Convert decimal 41 to binary.

	Integer Quotient		Remainder	Coefficient
41/2 =	20	+	½	$a0 = 1$
20/2 =	10	+	0	$a1 = 0$
10/2 =	5	+	0	$a2 = 0$
5/2 =	2	+	½	$a3 = 1$
2/2 =	1	+	0	$a4 = 0$
1/2 =	0	+	½	$a5 = 1$

Integer	Remainder
41	
20	1
10	0
5	0
2	1
1	0
0	1

Answer

=101001

answer : $(41)_{10} = (a5a4a3a2a1a0)2 = (101001)2$

Octal and Hexadecimal

#’s complement

- (r-1)’s complement of N is **(rn-1) - N**
  - n is equal to N’s digit

- r=10, r-1=9, the 9’s complements of N is (10n - 1) - N

Ex) the 9’s complements of 546700 is 999999 (= 1000000 – 1) -546700 = 453299

the 9’s complements of 012398 is 999999-012398 = 987601

- In the case of binary, r=2, r-1=1
  1’s complement of N is (2n - 1) - N

Ex) the 1’s complements of 1011000 is (10000000 – 1) - 1011000 = 1111111 – 1011000 = 0100111

the 1’s complements of 0101101 is 1010010

#'s complement

- rn-N = [(rn-1)-N] + 1
  - r's complements is equal to (r-1)'s complements + 1

Ex) the 2's complements of 1011000 is 0100111 + 1 = 0101000

the 2's complements of 0101101 is 1010010 + 1 = 1010011

#'s complement

- Ex1-7) X=1010100, Y=1000011, (a) X-Y, (b) Y-X

X =	1010100
2's complement of Y =	+0111101
Sum =	10010001
Discard end carry 2^7 =	-10000000
Answer: X-Y =	0010001

Y =	1000011
2's complement of X =	+0101100
Sum =	1101111

There is no carry.

The answer is Y-X = -(2's complement of 1101111)=-0010001

#'s complement

- Ex) X=1010100, Y=1010100, X-Y

X =	1010100
2's complement of Y =	+0101100
Sum =	10000000
Discard end carry 2^7 =	-10000000
Answer: X-Y =	0000000

The answer is X-Y = 0

Shift operation in binary

(11010)2 =1×24+1×23+0×22+1×21+0×20 = (26)10

Shift Left 2

(1101000)2 = 1 x 26 + 1 x 25 + 0 x 24 + 1 x 23 + 0 x 22 + 0 x 21 + 0 X 20

= (104)10 = (26)10 x 4

(110100)2 = 1×25+1×24+0×23+1×22+0×21+0×20 = (52)10

Shift Right 2

(1101)2 = 1×23+1×22+0×21+1×20 = (13)10 = (52)10 / 4

inst_data.mif

Byte Address

0
4
8
12…20
24
28
36
40
.
.
.

Word Index

```
000  :    EA000006;
001  :    EAFFFFFE;
002  :    EA0000A7;
[003..005]  :   EAFFFFFE;
006  :    EA0000A4;
007  :    EAFFFFFE;
008  :    E59F2EC8;
009  :    E3A00040;
00A  :    E5820010;
00B  :    E5820014;
00C  :    E5820018;
00D  :    E582001C;
00E  :    E5820020;
00F  :    E5820024;
010  :    E3A0003F;
011  :    E5820028;
012  :    E3A00008;
013  :    E582002C;
014  :    E59F3E9C;
015  :    E59F1E9C;
016  :    E5831000;
017  :    E59F9E98;
018  :    E3A08000;
019  :    E5898000;
01A  :    E5898004;
01B  :    E5898008;
01C  :    E589800C;
01D  :    E5898010;
01E  :    E5898014;
01F  :    E5898018;
020  :    E59FDE78;
021  :    E5931200;
022  :    E3510001;
023  :    0A000000;
024  :    EAFFFFFB;
```

EA000006

Instruction

Example

- **EA000006**

➢ Change instructions to binary format

- 1110 **101**0 0000 0000 0000 0000 0000 0110 (2)

➢ Translate the binary instructions to assembly codes by referring to the reference file

008

- B #[redacted] (**= 0000 0000 0000 0000 0000 0110)**;

➢ Describe what instruction means

- 1. Sign-extending the 24-bit signed (two’s complement) immediate to 30 bits
  - 0000 0000 0000 0000 0000 0110 -> **00 0000** 0000 0000 0000 0000 0000 0110
- 2. Shifting the result left two bits to form a 32-bit value
  - 0000 0000 0000 0000 0000 0000 0001 10**00** = 610 * 4 = 2410
- Adding this to the contents of the PC, which contains the address of the branch instruction plus 8 bytes.
  - Make ‘32’ by adding the current instruction address ‘(0*4)+8’ and ‘24’
  - Divide ‘32’ into 4 so that it branches at first among the word-unit instructions : 32 / 4 = 8
    - Next instruction will be E59F2EC8 at the address 008

Example

EAFFFFFE

Change instructions to binary format

1110 **1010** 1111 1111 1111 1111 1111 1110 (2)

Translate the binary instructions to assembly codes by referring to the reference file

B #[redacted] **(= 1111 1111 1111 1111 1111 1110)**;

(= 2's complement of 0000 0000 0000 0000 0000 0010)

Describe what instruction means

1. Sign-extending the 24-bit signed (two's complement) immediate to 30 bits

   1111 1111 1111 1111 1111 1110 -> **11 1111** 1111 1111 1111 1111 1111 1110

2. Shifting the result left two bits to form a 32-bit value

   1111 1111 1111 1111 1111 1111 11111 10**00** = -210 * 4 = -810

Adding this to the contents of the PC, which contains the address of the branch instruction plus 8 bytes.

Make '4' by adding the current instruction address '(1*4)+8' and '-8'

Divide '4' into 4 so that it branches at first among the word-unit instructions : 4/4 = 1

- Because it branches to the same instruction, the same instruction repeats indefinitely