COMPUTER ARCHITECTURE

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

LECTURE 4

BY NDUMISO E. KHUMALO

COMPUTER ARITHMETIC

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

Objective

- ALU
- Data Representation
- Signed and unsigned number
- arithmetic operation
- application of complements
- floating point representation

ALU

- The ALU is that part of the computer that actually performs arithmetic and logical operations on data

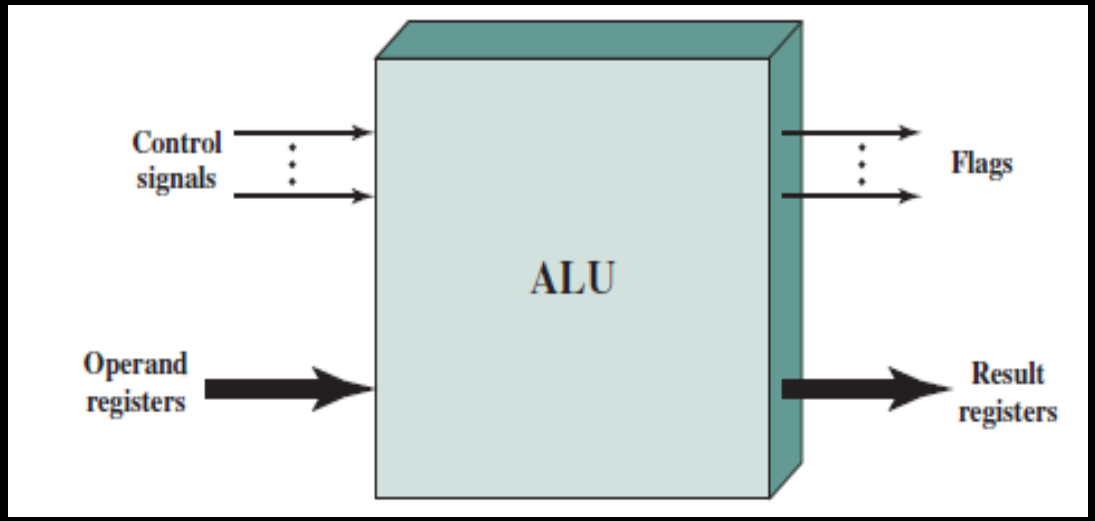
- All of the other elements of the computer system—control unit, registers, memory, I/O—are there mainly to bring data into the ALU for it to process and then to take the results back out

ALU

- An ALU and indeed, all electronic components in the computer, are based on the use of simple digital logic devices that can store binary digits and perform simple Boolean logic operations.
- Operands for arithmetic and logic operations are presented to the ALU in registers, and the results of an operation are stored in registers.
  - These registers are temporary storage locations within the processor that are connected by signal paths to the ALU

ALU

- The ALU may also set flags as the result of an operation.
  - E.g., an overflow flag is set to 1 if the result of a computation exceeds the length of the register into which it is to be stored.
  - ALU IMPUTS AND OUTPUTS



Number Representation

- In the binary number system, arbitrary numbers can be represented with just the digits zero and one, the minus sign (for negative numbers), and the period, or radix point (for numbers with a fractional component).
  - $-1101.0101_2 = -13.3125_{10}$
- For purposes of computer storage and processing, however, we do not have the benefit of special symbols for the minus sign and radix point

Number Representation

- Only binary digits (0 and 1) may be used to represent numbers.
- If we are limited to nonnegative integers, the representation is straightforward.
- An 8-bit word can represent the numbers from 0 to 255
- In fact an n-bit word can represent the numbers 0 to 2^n-1

Number Representation: Sign-Magnitude Representation

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

- There are several alternative conventions used to represent negative as well as positive integers, all of which involve treating the most significant (leftmost) bit in the word as a sign bit.
  - If the sign bit is 0, the number is positive; if the sign bit is 1, the number is negative.
- The simplest form of representation that employs a sign bit is the sign-magnitude representation.
  - In an n-bit word, the rightmost n - 1 bits hold the magnitude

Number Representation: Sign-Magnitude Representation

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

- Eg
  - +18 = 00010010
  - -18 = 10010010 (sign magnitude)

- There are several drawbacks to sign-magnitude representation
  - One is that addition and subtraction require a consideration of both the signs of the numbers & their relative magnitudes to carry out the required operation.

Number Representation: Sign-Magnitude Representation

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

- Another drawback is that there are two representations of 0:
  - + 0_{10} = 00000000
  - - 0_{10} = 10000000 (sign magnitude)

- Because of these drawbacks, sign-magnitude representation is rarely used in implementing the integer portion of the ALU.
- Instead, the most common scheme is twos complement representation.

Number Representation: Twos Complement Representation

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

- twos complement representation uses the most significant bit as a sign bit, making it easy to test whether an integer is positive or negative

- It differs from the use of the sign-magnitude representation in the way that the other bits are interpreted.

Number Representation: Twos Complement Representation

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

- Characteristics of Twos Complement Representation and Arithmetic

Range	-2^{n-1} through $2^{n-1} - 1$
Number of Representations of Zero	One
Negation	Take the Boolean complement of each bit of the corresponding positive number, then add 1 to the resulting bit pattern viewed as an unsigned integer.
Expansion of Bit Length	Add additional bit positions to the left and fill in with the value of the original sign bit.
Overflow Rule	If two numbers with the same sign (both positive or both negative) are added, then overflow occurs if and only if the result has the opposite sign.
Subtraction Rule	To subtract B from A, take the twos complement of B and add it to A.

Number Representation: Twos Complement Representation

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

- Consider an n-bit integer, A, in twos complement representation.
  - If A is positive, then the sign bit, a_{n-1}, is zero.
  - The remaining bits represent the magnitude of the number in the same fashion as for sign magnitude
  - The number zero is identified as positive and therefore has a 0 sign bit and a magnitude of all 0s.

Number Representation: Twos Complement Representation

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

- We can see that the range of positive integers that may be represented is from 0 (all of the magnitude bits are 0) through $2^{n-1} - 1$ (all of the magnitude bits are 1).
  - Any larger number would require more bits.
- Now, for a negative number A(A 6 0), the sign bit, a_{n-1}, is one.
  - The remaining n - 1 bits can take on any one of 2^{n-1} values.

Number Representation: Twos Complement Representation

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

- Therefore, the range of negative integers that can be represented is from -1 to -2^{n-1}.
- We would like to assign the bit values to negative integers in such a way that arithmetic can be handled in a straightforward fashion, similar to unsigned integer arithmetic.
  - In unsigned integer representation, to compute the value of an integer from the bit representation, the weight of the most significant bit is $+2^{n-1}$

Number Representation: Twos Complement Representation

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

- What about if the weight of the most significant bit is -2^{n-1}?

Number Representation: Integer Arithmetic

LIMKOKWING UNIVERSITY —COLLEGE— OF CREATIVE TECHNOLOGY

- In sign-magnitude representation, the rule for forming the negation of an integer is simple: invert the sign bit.
- In twos complement notation, the negation of an integer can be formed with the following rules
  - Take the Boolean complement of each bit of the integer (including the sign bit). That is, set each 1 to 0 and each 0 to 1.
  - Treating the result as an unsigned binary integer, add

Number Representation: Integer Arithmetic

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

- This two-step process is referred to as the twos complement operation, or the taking of the twos complement of an integer
  - +18 = 00010010 (twos complement)

    bitwise complement = 11101101 + 1 =11101110 = **-**18
  - As expected, the negative of the negative of that number is itself:
    - **-**18 = 11101110 (twos complement)

    bitwise complement = 00010001+ 1 = 00010010 = +18

Number Representation: Integer Arithmetic

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

- Addition
  - Addition proceeds as if the two numbers were unsigned integers
  - If the result of the operation is positive, we get a positive number in twos complement form, which is the same as in unsigned-integer form
  - If the result of the operation is negative, we get a negative number in twos complement form
    - Note that, in some instances, there is a carry bit beyond the end of the word, which is ignored.

Number Representation: Integer Arithmetic

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

- Addition
  - On any addition, the result may be larger than can be held in the word size being used.
  - This condition is called overflow.
    - When overflow occurs, the ALU must signal this fact so that no attempt is made to use the result
  - To detect overflow, the following rule is observed:
    - overflow rule: If two numbers are added, and they are both positive or both negative, then overflow occurs if and only if the result has the opposite sign.

Number Representation: Integer Arithmetic

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

- Addition of Numbers in Twos Complement Representation

1001 = −7 +0101 = 5 1110 = −2 (a) (−7) + (+5)	1100 = −4 +0100 = 4 10000 = 0 (b) (−4) + (+4)
0011 = 3 +0100 = 4 0111 = 7 (c) (+3) + (+4)	1100 = −4 +1111 = −1 11011 = −5 (d) (−4) + (−1)
0101 = 5 +0100 = 4 1001 = Overflow (e) (+5) + (+4)	1001 = −7 +1010 = −6 10011 = Overflow (f) (−7) + (−6)

Number Representation: Integer Arithmetic

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

- SUBTRACTION
  - subtraction rule: To subtract one number (subtrahend) from another(minuend), take the twos complement (negation) of the subtrahend and add it to the minuend.

Number Representation: Integer Arithmetic

LIMKOKWING UNIVERSITY —COLLEGE— OF CREATIVE TECHNOLOGY

- Subtraction of Numbers in Twos Complement Representation (M - S)

```
          0010  =   2
        +1001   =  −7
          1011  =  −5

(a)  M  =  2  =  0010
     S  =  7  =  0111
    −S  =      1001
```

```
          0101  =   5
        +1110   =  −2
         10011  =   3

(b)  M  =  5  =  0101
     S  =  2  =  0010
    −S  =      1110
```

```
          1011  =  −5
        +1110   =  −2
         11001  =  −7

(c)  M  =  −5  =  1011
     S  =   2  =  0010
    −S  =      1110
```

```
          0101  =  5
        +0010   =  2
          0111  =  7

(d)  M  =   5  =  0101
     S  =  −2  =  1110
    −S  =      0010
```

```
          0111  =  7
        +0111   =  7
          1110  =  Overflow

(e)  M  =   7  =  0111
     S  =  −7  =  1001
    −S  =      0111
```

```
          1010  =  −6
        +1100   =  −4
         10110  =  Overflow

(f)  M  =  −6  =  1010
     S  =   4  =  0100
    −S  =      1100
```

Number Representation: Integer Arithmetic

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

- MULTIPLICATION
  - Compared with addition and subtraction, multiplication is a complex operation, whether performed in hardware or software
  - the multiplication of unsigned binary integers, as might be carried out using paper and pencil. Several important observations can be made:
    - 1. Multiplication involves the generation of partial products, one for each digit in the multiplier. These partial products are then summed to produce the final product.

Number Representation: Integer Arithmetic

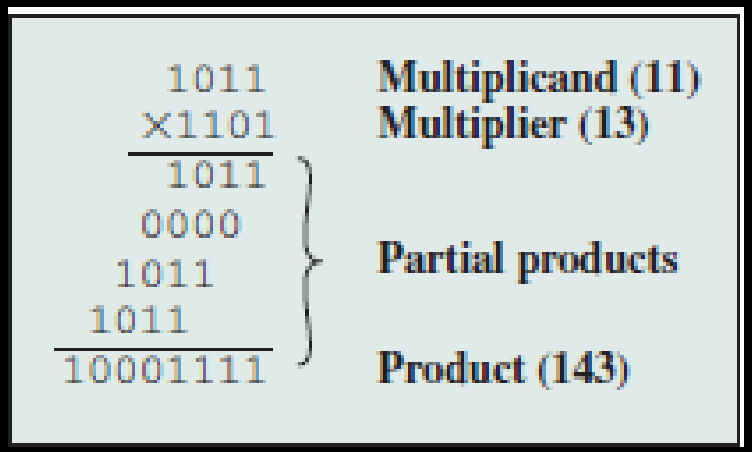
LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

- MULTIPLICATION
  - 2. The partial products are easily defined. When the multiplier bit is 0, the partial product is 0. When the multiplier is 1, the partial product is the multiplicand.
  - 3. The total product is produced by summing the partial products. For this operation, each successive partial product is shifted one position to the left relative to the preceding partial product.
  - 4. The multiplication of two n-bit binary integers results in a product of up to 2^n bits in length (e.g., 11 * 11 = 1001).

Number Representation: Integer Arithmetic

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

- Multiplication of Unsigned Binary Integers



  - Compared with the pencil-and-paper approach, there are several things we can do to make computerized multiplication more efficient.

Number Representation: Integer Arithmetic

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

- First, we can perform a running addition on the partial products rather than waiting until the end. This eliminates the need for storage of all the partial products; fewer registers are needed
- Second, we an save some time on the generation of partial products. For each 1 on the multiplier, an add and a shift operation are required; but for each 0, only a shift is required
- As shown in diagam

Number Representation: Integer Arithmetic

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

—

C	A	Q	M		
0	0000	1101	1011	Initial values	
0	1011	1101	1011	Add	First cycle
0	0101	1110	1011	Shift	
0	0010	1111	1011	Shift	Second cycle
0	1101	1111	1011	Add	Third cycle
0	0110	1111	1011	Shift	
1	0001	1111	1011	Add	Fourth cycle
0	1000	1111	1011	Shift	

Number Representation: Integer Arithmetic

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

- The multiplier and multiplicand are loaded into two registers (Q and M).
- A third register, the A register, is also needed and is initially set to 0.
- There is also a 1-bit C register, initialized to 0, which holds a potential carry bit resulting from addition.

Number Representation: Integer Arithmetic

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

- The operation of the multiplier is as follows:
  - Control logic reads the bits of the multiplier one at a time
  - If Q_0 is 1, then the multiplicand is added to the A register and the result is stored in the A register, with the C bit used for overflow.
  - Then all of the bits of the C, A, & Q registers are shifted to the right one bit, so that the C bit goes into A_{n-1}, A_0 goes into Q_{n-1}, and Q_0 is lost.

Number Representation: Integer Arithmetic

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

- If Q0 is 0, then no addition is performed, just the shift
- This process is repeated for each bit of the original multiplier.
- The resulting 2n-bit product is contained in the A and Q registers

- Twos complement multiplication
  - We have seen that addition and subtraction can be performed on numbers in twos complement notation by treating them as unsigned integers

Number Representation: Integer Arithmetic

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

- Twos complement multiplication
  - We have seen that addition and subtraction can be performed on numbers in twos complement notation by treating them as unsigned integers
  - Consider 1001 + 0011 = 1100
    - If these numbers are considered to be unsigned integers, then we are adding 9 (1001) plus 3 (0011) to get 12 (1100).
    - As twos complement integers, we are adding - 7(1001) to 3 (0011) to get - 4(1100).

Number Representation: Integer Arithmetic

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

- Unfortunately, this will not work for multiplication
  - Previously, we multiplied 11 (1011) by 13 (1101) to get 143 (10001111).
    - If we interpret these as twos complement numbers, we have -5(1011) times - 3 (1101) equals - 113 (10001111)
  - This example demonstrates that straightforward multiplication will not work if both the multiplicand and multiplier are negative.
    - In fact, it will not work if either the multiplicand or the multiplier is negative

Number Representation: Integer Arithmetic

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

- There are a number of ways out of this dilemma.
  - One would be to convert both multiplier and multiplicand to positive numbers, perform the multiplication, and then take the twos complement of the result if and only if the sign of the two original numbers differed

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

Floating point representation

- With a fixed-point notation (e.g., twos complement) it is possible to represent a range of positive and negative integers centered on or near 0.

- By assuming a fixed binary or radix point, this format allows the representation of numbers with a fractional component as well

Floating point representation

- This approach has limitations.
  - Very large numbers cannot be represented, nor can very small fractions.
  - Furthermore, the fractional part of the quotient in a division of two large numbers could be lost
  - For decimal numbers, we get around this limitation by using scientific notation.
    - Thus, 976,000,000,000,000 can be represented as $9.76 * 10^{14}$, and 0.0000000000000976 can be represented as $9.76 * 10^{-14}$

Floating point representation

- What we have done, in effect, is dynamically to slide the decimal point to a convenient location and use the exponent of 10 to keep track of that decimal point.
  - This allows a range of very large and very small numbers to be represented with only a few digits.
- This same approach can be taken with binary numbers.
  - We can represent a number in the form $\pm S \times B^{\pm E}$

Floating point representation

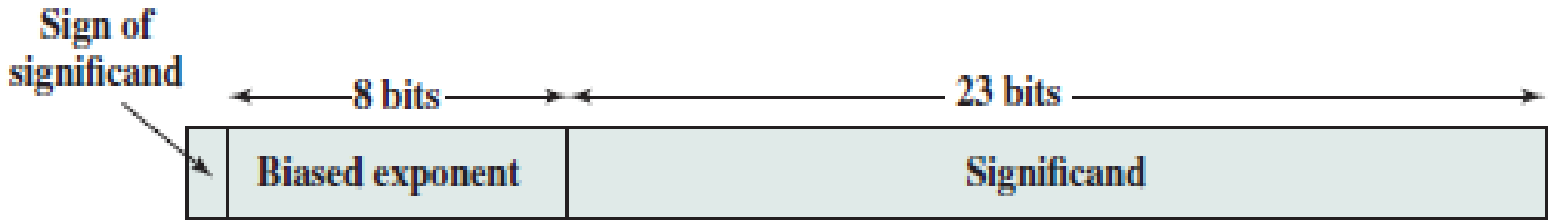
- This number can be stored in a binary word with three fields:
  - Sign: plus or minus
  - Significand S
  - Exponent E
- The base B is implicit and need not be stored because it is the same for all numbers.

Floating point representation

- Typically, it is assumed that the radix point is to the right of the leftmost, or most significant, bit of the significand
  - That is, there is one bit to the left of the radix point

Floating point representation

- Typical 32-Bit Floating-Point Format



(a) Format

$1.1010001 \times 2^{10100} = 0\ 10010011\ 10100010000000000000000 = 1.6328125 \times 2^{20}$

$-1.1010001 \times 2^{10100} = 1\ 10010011\ 10100010000000000000000 = -1.6328125 \times 2^{20}$

$1.1010001 \times 2^{-10100} = 0\ 01101011\ 10100010000000000000000 = 1.6328125 \times 2^{-20}$

$-1.1010001 \times 2^{-10100} = 1\ 01101011\ 10100010000000000000000 = -1.6328125 \times 2^{-20}$

Floating point representation

- Figure shows a typical 32-bit floating-point format.
  - The leftmost bit stores the sign of the number (0 = positive, 1 = negative).
  - The exponent value is stored in the next 8 bits. The representation used is known as a biased representation.
  - A fixed value, called the bias, is subtracted from the field to get the true exponent value
  - Typically, the bias equals $(2^{k-1} - 1)$, where k is the number of bits in the binary exponent

Floating point representation

- Figure shows a typical 32-bit floating-point format.
  - In this case, the 8-bit field yields the numbers 0 through 255.
  - With a bias of 127 (2^7 - 1), the true exponent values are in the range -127 to + 128.
  - In this example, the base is assumed to be 2
  - The final portion of the word (23 bits in this case) is the significand

Floating point representation

- Any floating-point number can be expressed in many ways.
  - The following are equivalent, where the significand is expressed in binary form:
    - $0.110 * 2^5$
    - $110 * 2^2$
    - $0.0110 * 2^6$

Floating point representation

- To simplify operations on floating-point numbers, it is typically required that they be normalized.
  - A normal number is one in which the most significant digit of the significand is nonzero.
  - For base 2 representation, a normal number is therefore one in which the most significant bit of the significand is one.

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

THE END

- NEXT TOPIC: INSTRUCTION SETS CHARACTERISTICS AND FUNCTIONS, ADDRESSING MODES AND FORMARTS