CSC231 - Assembly

Week #6

Dominique Thiébaut
dthiebaut@smith.edu
Misc. Notes & Review
You Can Add A Register to Itself

; $x = 2 \times (a-b)$

    mov eax, dword[a] ; eax <- a
    sub eax, dword[b] ; eax <- a - b
    add eax, eax      ; eax <- eax + eax
    mov dword[x], eax ; x <- 2*(a-b)
Speed of Computation
Frequency & Cycle Time
Frequency: 3.2 GHz

cycle: $\frac{1}{3.2 \text{ GHz}} = 0.3125 \text{ ns}$

sec
ms
us
ns
#include "./m328Pdef.inc"

ldi r16,0b00100000
out DDRB,r16
out PortB,r16
Start:
rjmp Start

Clock speed: 16 MHz
~1/200 speed of Pentium

Raspberry Pi

Clock speed: 1.4 GHz
~1/3 speed of Pentium

/* -- first.s */
/* This is a comment */
.global main /* 'main' is our entry point and must be global */
main:          /* This is main */
    mov r0, #2 /* Put a 2 inside the register r0 */
    bx lr     /* Return from main */

N-Queens Problem

Let’s go to Boston this afternoon. I’ll drive!
Afternoon Plans?

Let’s go to Boston this afternoon. I’ll drive!
Afternoon Plans?

Let's go for a drive in the woods. I’ll drive!
Know your toolbox!
Summit: Fastest Supercomputer
Oak Ridge National Laboratory

https://www.olcf.ornl.gov/summit/
NUMBER SYSTEMS
Decimal

- Number of digits, the base
- Count in decimal
- Express number as sum of products
- Add two digits
- Add two numbers
Binary

- Number of digits, the base
- Count in binary
- Express number as sum of products
- Add two digits
- Add two numbers
• Number of digits, the base
• Count in base 3
• Express number as sum of products
• Add two digits
• Add two numbers
Hexadecimal

- Number of digits, the base
- Count in hex
- Express number as sum of products
- Add two digits
- Add two numbers
Conversion From One Base To Another
Decimal to Binary
# prompts user for an integer
# decomposes the integer into binary

x = int( input( "> " ) )
binary = ""

while True:
    if x==0:
        break
    remainder = x % 2
    quotient  = x // 2
    if remainder == 0:
        binary = "0" + binary
    else:
        binary = "1" + binary

    print( "%5d = %5d * 2 + %d    quotient=%5d remainder=%d binary=%16s" 
           % (x, quotient, remainder, quotient, remainder, binary ) )
    x = quotient
# prompts user for an integer
# decomposes the integer into binary

x = int(input("> "))
binary = ""

while True:
    if x==0:
        break
    remainder = x % 2
    quotient  = x // 2
    if remainder == 0:
        binary = "0" + binary
    else:
        binary = "1" + binary
    print("%5d = %5d * 2 + %d    quotient=%5d remainder=%d binary=%16s" % (x, quotient, remainder, quotient, remainder, binary))
    x = quotient
<table>
<thead>
<tr>
<th>Binary</th>
<th>Hexadecimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>0</td>
</tr>
<tr>
<td>0001</td>
<td>1</td>
</tr>
<tr>
<td>0010</td>
<td>2</td>
</tr>
<tr>
<td>0011</td>
<td>3</td>
</tr>
<tr>
<td>0100</td>
<td>4</td>
</tr>
<tr>
<td>0101</td>
<td>5</td>
</tr>
<tr>
<td>0110</td>
<td>6</td>
</tr>
<tr>
<td>0111</td>
<td>7</td>
</tr>
<tr>
<td>1000</td>
<td>8</td>
</tr>
<tr>
<td>1001</td>
<td>9</td>
</tr>
<tr>
<td>1010</td>
<td>A</td>
</tr>
<tr>
<td>1011</td>
<td>B</td>
</tr>
<tr>
<td>1100</td>
<td>C</td>
</tr>
<tr>
<td>1101</td>
<td>D</td>
</tr>
<tr>
<td>1110</td>
<td>E</td>
</tr>
<tr>
<td>1111</td>
<td>F</td>
</tr>
</tbody>
</table>
Hexadecimal to Binary

<table>
<thead>
<tr>
<th>Hexadecimal</th>
<th>Binary</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>0</td>
</tr>
<tr>
<td>0001</td>
<td>1</td>
</tr>
<tr>
<td>0010</td>
<td>2</td>
</tr>
<tr>
<td>0011</td>
<td>3</td>
</tr>
<tr>
<td>0100</td>
<td>4</td>
</tr>
<tr>
<td>0101</td>
<td>5</td>
</tr>
<tr>
<td>0110</td>
<td>6</td>
</tr>
<tr>
<td>0111</td>
<td>7</td>
</tr>
<tr>
<td>1000</td>
<td>8</td>
</tr>
<tr>
<td>1001</td>
<td>9</td>
</tr>
<tr>
<td>1010</td>
<td>A</td>
</tr>
<tr>
<td>1011</td>
<td>B</td>
</tr>
<tr>
<td>1100</td>
<td>C</td>
</tr>
<tr>
<td>1101</td>
<td>D</td>
</tr>
<tr>
<td>1110</td>
<td>E</td>
</tr>
<tr>
<td>1111</td>
<td>F</td>
</tr>
</tbody>
</table>
Powers of 16: $16^n$

- $16^0 = 1$
- $16^1 = 16$
- $16^2 = 256$
- $16^3 = 4096$
- $16^4 = 65536$
- $16^5 = 1048576$
- $16^6 = 16777216$
- $16^7 = 268435456$
- $16^8 = 4294967296$
- $16^9 = 68719476736$
- $16^{10} = 1099511627776$
- $16^{11} = 17592186044416$
- $16^{12} = 281474976710656$
- $16^{13} = 4503599627370496$
Decimal to Hexadecimal
Exercises