Test 1 Study Guide

October 4, 2018

Direct Mapping Consider a machine with a byte addressable main memory of 2 24 bytes and a block size of 32 bytes. Assume that a direct mapped cache consisting of 64 lines is used with this machine. How is a 24-bit memory address divided into tag, line/set, and word/offset? How many total bytes of memory can be stored in the cache? Suppose the byte with address (1101 1011 0100 1110 1010 0111) 2 is stored in the cache. A number of other bytes are stored with this byte on the same line of cache. What is the highest address number in this group?

Solution Block size = 32 bytes
Size of Block/Word = $\log_2 32 = 5$ bytes
64 lines $\rightarrow \log_2 64 = 6 bytes$
\# of lines = $\frac{Cachesize}{BlockSize} \rightarrow 64 = \frac{CacheSize}{32bytes}$
$\rightarrow 64 * 32bytes = CacheSize = 2KBytes$
Memory block = $24 = tag + 6bytesline/set + 5word/offset$
$\rightarrow 24 - (6+5) = 13bytes = tag$
Dividing 110110110100111010100111

13Byte Tag	6Bytes Word	5 Bytes word
1101101101001	110101	00111

The highest addressable is $2^{blocksize} = 2^{32}$

Booths Method AKA multiplying Two's Compliment Binary Numbers If at first you don't get the same numbers switch the two numbers

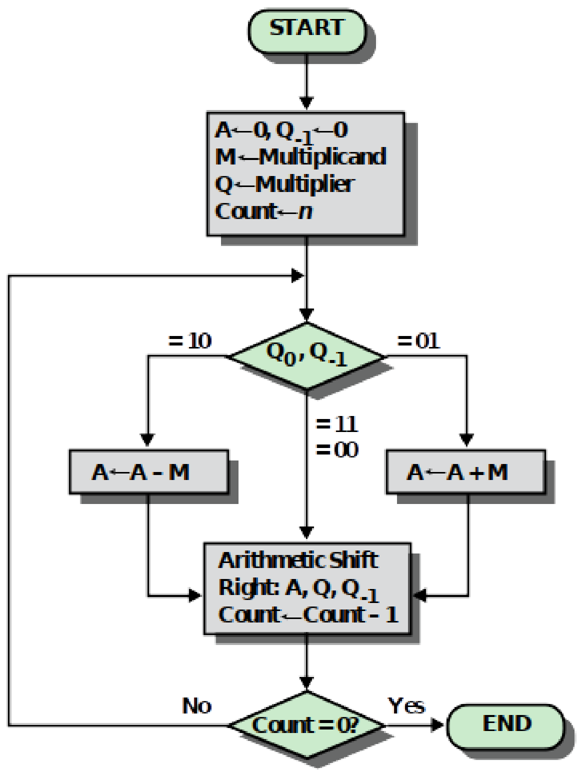

Figure 10.12 Booth's Algorithm for Twos Complement Multiplication

Instructions Per Seconds Assume the following instruction mix for a given benchmark program: 1535million instructions and that load instructions require two cycles, branches require 4 cycles, integer addition and store instructions require 8 cycles and integer multiplies require 16 cycles, compute the overall cycles-per-instruction or CPI. Determine the Millions of Instructions per Second (MIPS) rate for a 3200-MHz processor. What is the execution time for this program?

Solution $CPI = \sum_{i=1}^{n} \frac{CPI_i x I_i}{I_c}$
Instruction Percentage * Inscrution # Cycles
$= .15 * 8 + .25 * 2 + .35 * 4 + .05 * 8 + .2 * 16 = 6.7$ cycles per second
$MIPS = \frac{Frequency}{CPI * 10^6} \rightarrow \frac{3200*10^6}{6.7*10^6} = \frac{3200}{6.7}$
= 4.77.16 Million Instructions Per Second
Total Execution Time $= I_c * CPI * \frac{1}{F}$
$10^6 * 6.7 * \frac{1}{3200*10^6}$
$= 2.1ms$

Sum of Products Stuff

A	B	C	F	SOP
0	0	0	0	0
0	0	1	0	0
0	1	0	1	$\overline{A}B\overline{C}$
0	1	1	1	$\overline{A}BC$
1	0	0	1	$A\overline{B} * \overline{C}$
1	0	1	1	$A\overline{B}C$
1	1	0	0	0
1	1	1	0	0

SOP $= \overline{A}B\overline{C} + \overline{A}BC + A\overline{B} * \overline{C} + A\overline{B}C$
POS $= (A+B+C)+(A+B+\overline{C})+(A+\overline{B}+C)+(A+\overline{B}+\overline{C})+(\overline{A}+B+C)+(\overline{A}+B+\overline{C})+(\overline{A}+\overline{B}+C)+(\overline{A}+\overline{B}+\overline{C})$ See notes for the rest

Caching Stuff A 64KB cache has 32 Byte blocks. Addresses are 32 bits. a) How many bits are used for the tag, index, and offset if the cache is direct-mapped? b) How would the address be divided if the cache were 2-way set associative instead? c) How many bits is the index for a fully associative cache. Explain your answers.

Solution 1)

Number of blocks = Cache-Size/Block-Size = 64 KB / 32 Bytes = 211
Number of blocks = 211 = 211
Tag + index + offset = 32
Tag + 11 + 5 = 32
Tag = 16

2)
Number of blocks = Cache-Size/Block-Size = 64 KB / 32 Bytes = 211
Number of Sets = 211 / 2 = 210
Tag + Set offset + Byte offset = 32
Tag + 10 + 5 = 32
Tag = 17

3) offset = log(32) = 5 bits
tag = 32-5 = 27 bits

Anotha One A 128 KB L1 cache has a 64 byte block size and is 4-way set-associative. How many sets does the cache have? How many bits are used for the offset, index, and tag, assuming that the CPU provides 32-bit addresses? How large is the tag array? Please show your steps.

Cache Size = 128 KB = 2^{7+10} B = 2^{17} B.

Block Size = 64 B

# of Blocks in cache = $\frac{2^{17} B}{2^{6} B} = 2^{11}$ Blocks.

# of sets = $\frac{2^{11} \text{ Blocks}}{4} = 2^{9}$

Tag size = 9

Block offset = $64 = 2^{6}$ = 6 bit

Tag	Index	Block offset
17 Bits	9 bits	6 bits

Tag size = $(32 - (9+6)) = 32 - 15 = 17$ bits.

Tag size = 17 bits

Tag	index	Block offset
17	9	6

Solution