Cheatography

STY 2013 Final Cheat Sheet

by [deleted] via cheatography.com/5328/cs/965/

assembly: registers and flags

%eax | %ecx | %edx | %ebx | %esi | %edx	Temporary data, General purpose registers
%esp | %ebp	Current: stack top | stack frame
%eip	Instruction pointer
CF | ZF | SF | OF	Carry flag | Zero flag | Sign flag | Overflow flag

Note: flags are not set by lea instruction.

assembly: jumps and shifts

sal | sar	arithmetic shift left | right
shl | shr	logical shift left | right
jz | jnz	jump if == 0,"zero" | != 0,"not zero"
je | jne | jg | jge | jl | jle	jump if == | != | > | >= | < | <=
js	jump and store
jmp | jmp *reg	unconditional relative jump| absolute jump, reg is a registry.
ja | jb (unsigned)	jump above | below

assembly: compares and flags

cmp b, a	a - b
test b, a	a & b
zf "zero flag"	set when a&b== 0
sf "signed flag"	set when a&b < 0

assembly: getting setting

lea a, b	load effective address a into b
mov a, b	move contents of a into b
cmov (z,nz,e,ne,g,ge,l,le,ng, nge,nl,nle,a,b, ...)	compare and move if condition is met.
movl %edx, %eax	eax = edx, eax bendir á edx
movl (%edx), %eax	eax = *edx, eax verður bendir á innihald edx
movl %edx, (%eax)	*eax = edx, eax bendir á bendinn að innihaldi edx
movl (%edx), (%eax)	*eax* = edx, yfirskrifar innihald eax með innihaldi edx.

Array shizznit

address(mn[i][j]) = 0+i*N+4j
address(nm[i][j]) = 0+i*M+4j

Given the arrays:
int mn[M][N]; and int nm[N][M];

Reading a disk sector(sequence)

1: CPU initiates disk read, writes cmd, lbn and desk to a DC port(address)

2: DC reads sector and performs a DMA transfer into main memory

3: DC notifies CPU with *interrupt* signal when DMA transfer completes

DC: Disk controller
DMA: direct memory access
cmd: command
lbn: logical block number
dest: destination

Memory system parameters

$N=2^n$	Number of addresses in virtual address space
$M=2^m$	Number of addresses in physical address space
$P=2^p$	Page size(bytes)

Components of PA(physical address)

PPO	Physical page offset(same as VPO)
PPN	Physical page number.
CO	Byte offset within cache line
CI	Cache index
CT	Cache tag

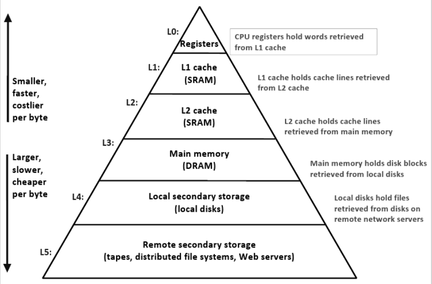
Components of VA(Virtual Address)

TLBI	TLB index
TLBT	TLB tag
VPO	Virtual page offset
VPN	Virtual page number

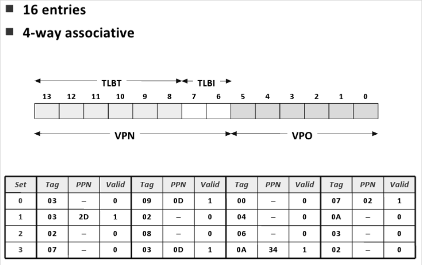
Locality

Temporal locality:
> Recently referenced items are likely to be referenced again in the near future.
Spatial locality:
> Items with nearby addresses tend to be referenced close together in time.

Memory Hierarchy


L0:
Registers
CPU registers hold words retrieved from L1 cache
L1:
L1 cache (SRAM)
L1 cache holds cache lines retrieved from L2 cache
L2:
L2 cache (SRAM)
L2 cache holds cache lines retrieved from main memory
L3:
Main memory (DRAM)
Main memory holds disk blocks retrieved from local disks
L4:
Local secondary storage (local disks)
Local disks hold files retrieved from disks on remote network servers
L5:
Remote secondary storage (tapes, distributed file systems, Web servers)
Smaller, faster, costlier per byte
Larger, slower, cheaper per byte

Simple Memory System TLB

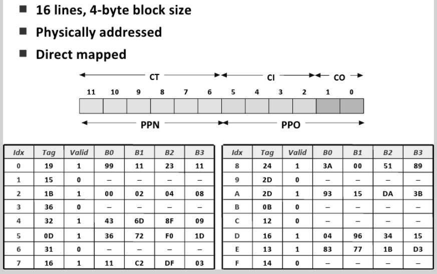


- 16 entries
- 4-way associative

TLBT
TLBI
13 12 11 10 9 8 7 6 5 4 3 2 1 0
VPN
VPO

Set	Tag	PPN	Valid	Tag	PPN	Valid	Tag	PPN	Valid	Tag	PPN	Valid
0	03	–	0	09	0D	1	00	–	0	07	02	1
1	03	2D	1	02	–	0	04	–	0	0A	–	0
2	02	–	0	08	–	0	06	–	0	03	–	0
3	07	–	0	03	0D	1	0A	34	1	02	–	0

Simple Memory System Cache



- 16 lines, 4-byte block size
- Physically addressed
- Direct mapped

CT
CI
CO
11 10 9 8 7 6 5 4 3 2 1 0
PPN
PPO

Idx	Tag	Valid	B0	B1	B2	B3
0	19	1	99	11	23	11
1	15	0	–	–	–	–
2	1B	1	00	02	04	08
3	36	0	–	–	–	–
4	32	1	43	6D	8F	09
5	0D	1	36	72	F0	1D
6	31	0	–	–	–	–
7	16	1	11	C2	DF	03

Idx	Tag	Valid	B0	B1	B2	B3
8	24	1	3A	00	51	89
9	2D	0	–	–	–	–
A	2D	1	93	15	DA	3B
B	0B	0	–	–	–	–
C	12	0	–	–	–	–
D	16	1	04	96	34	15
E	13	1	83	77	1B	D3
F	14	0	–	–	–	–

Cache

TLB	holds recently used PTE's, located on the cpu chip.
PTE	Page table entry, physical address of data in cache/memory

Signals

ID	Name	Default Action	Event
2	SIGINT	Terminate	Interupt,ctrl-c
9	SIGKILL	Terminate	Kill (unavoidable)
11	SIGSEGV	Terminate&Dump	Segfault
14	SIGALRM	Terminate	Timer signal
15	SIGTERM	Terminate	Kill nicely(catchable)
17	SIGCHLD	Ignore	Child stoppd or killd