Memory Organization

Viewed as a large, single-dimension array, with an address.

A memory address is an index into the array

"Byte addressing" means that the index points to a byte of memory.



• • •

Memory Organization (cont.)

Bytes are nice, but most data items use larger "words" For MIPS, a word is 32 bits or 4 bytes.



Registers hold 32 bits of data

 2^{32} bytes with byte addresses from 0 to 2^{32} -1

 2^{30} words with byte addresses 0, 4, 8, ... 2^{32} -4

Words are aligned

i.e., what are the least 2 significant bits of a word address?

Addressing Objects: Endianess and Alignment

Big Endian: address of most significant byte = word address (xx00 = Big End of word)IBM 360/370, Motorola 68k, MIPS, Sparc, HP PA address of least significant byte = word address Little Endian: (xx00 = Little End of word)Intel 80x86, DEC Vax, DEC Alpha (Windows NT) little endian byte 0 3 2 0 1 msb lsb 2 3 1 0 2 3 Ο Aligned big endian byte 0 Alignment: require that objects fall on address that is multiple of their size. Not Aligned

90

Data Storage

Characters: 8 bits (byte) Integers: 32 bits (word) Array: Sequence of locations Pointer: Address

000	- 11		
001	•		
002			
003			
004	258		
005	9		
006	Θ		
007	9	Γ	
008		T	,
009	deal.		
010	KO		
011			•
012		١	-
013	int		
014			
015	40	1	



Loads & Stores



Note: Ibu & sb load & store bytes