Computer Science 210

Computer Systems 1

2007 Semester 1

Lecture Notes

Load/Store Instructions

Lecture 5
24 Mar 07

James Goodman

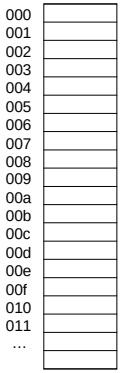
Department of Computer Science

Recommended Readings

- Today's lecture mostly based on Section 4.2 of Dr. Hutton's notes.

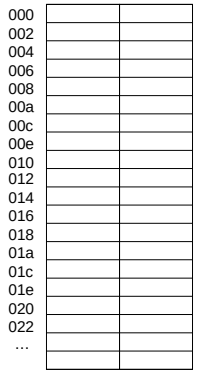
Views of Memory

- An array of bytes



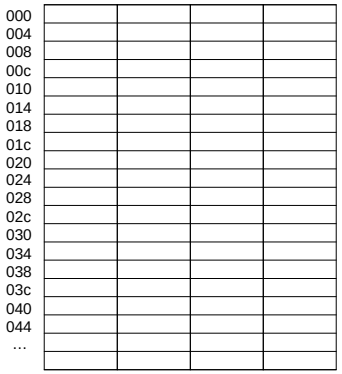
Views of Memory

- An array of words



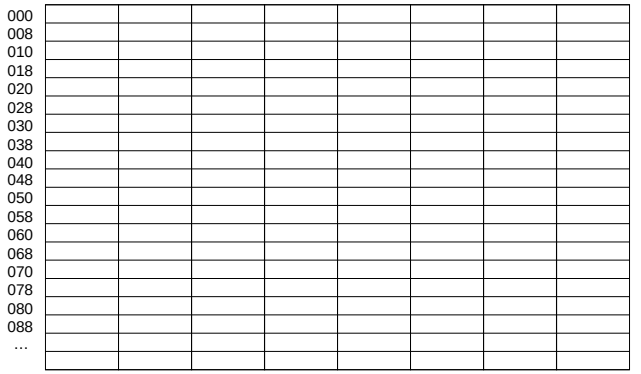
Views of Memory

- An array of longwords

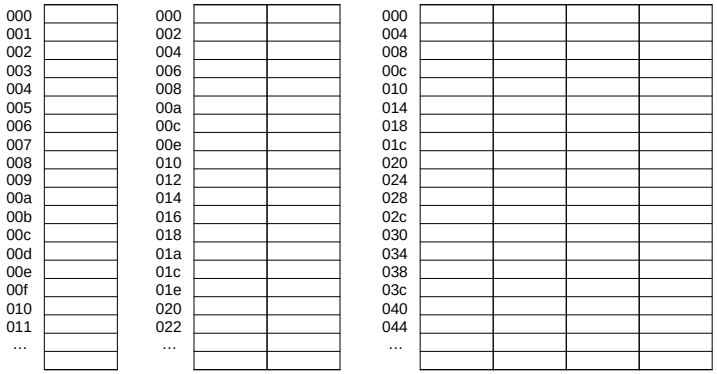


Views of Memory

- An array of quadwords

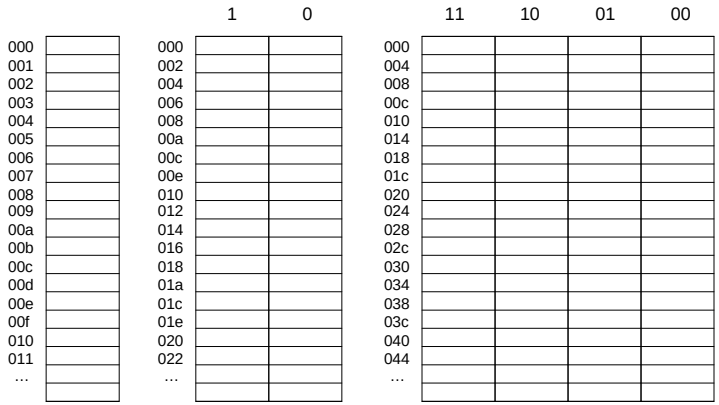


Which is Correct View of Memory?



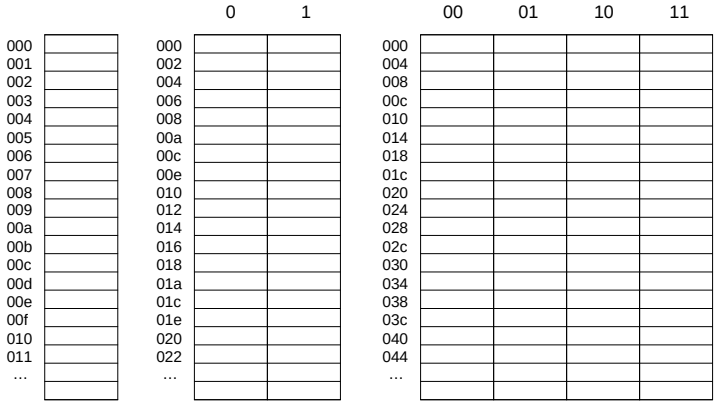
Byte Order

- Little Endian



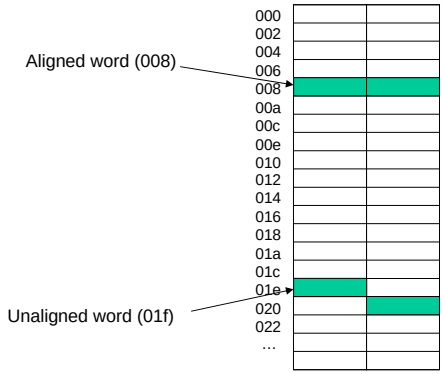
Byte Order

- Big Endian



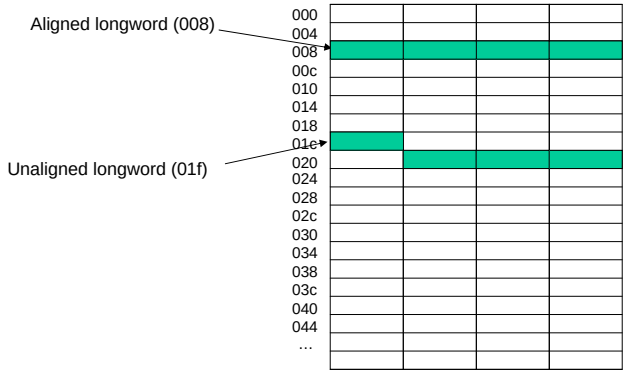
Alignment

- An array of words (little-endian)



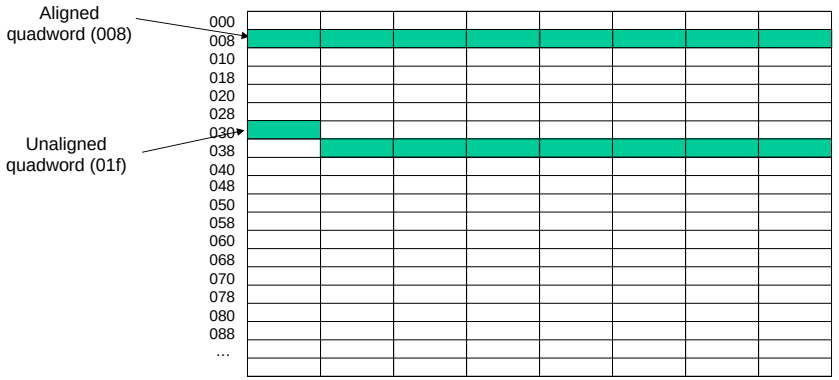
Views of Memory

- An array of longwords (little-endian)



Views of Memory

- An array of quadwords (little-endian)



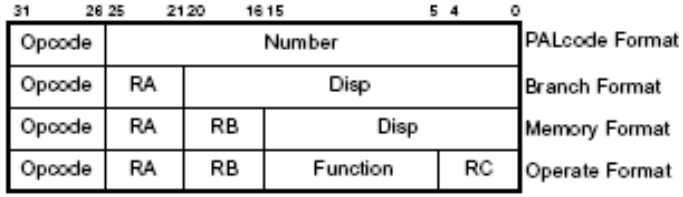
Load Instructions

- **`ldq reg, …`** Load quadword
- **`ldl reg, …`** Load *sign-extended* longword
- **`ldwu reg, …`** Load *zero-extended* word
- **`ldbu reg, …`** Load *zero-extended* byte

- **`stq reg, …`** Store quadword
- **`stl reg, …`** Store longword
- **`stw reg, …`** Store word
- **`stb reg, …`** Store byte

Motivation: Instruction Formats

Figure 1–1: Instruction Format Overview



– From The Alpha Architecture Handbook, Compaq Computer Corporation, 1998.

Load & Stores

```
ldq Reg, Address  ! Direct
```

What's the problem?

- Address is stored as a constant inside the instruction
  - How to *dynamically* change the address?
- Instructions should be small, fixed size
  - Addresses are large
  - How to store a 51-bit address in a 32-bit instruction?
- Also a problem for branch instructions:

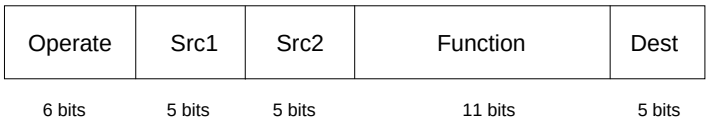
```
bne Reg, Address
```


Where Do the Bits Go?

Opcode	Src1	Src2	Dest

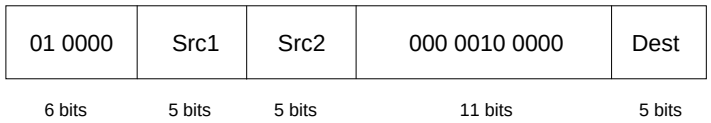
- Register specification requires 5 bits
- Memory specification requires up to 51 bits (at least 43)
- Opcode specification requires ?
  - 515 unique opcodes
  - 10 bits required?

Arithmetic/Logical Instruction



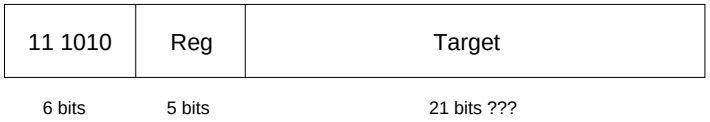
- Opcode indicates a class of instructions
  - Opcode 10_{16} indicates an arithmetic function
  - Opcode 11_{16} indicates a logical function
  - Opcode 12_{16} indicates a shift function
- "Function" is extended Opcode, specifying which arithmetic/logical function

Example: addq



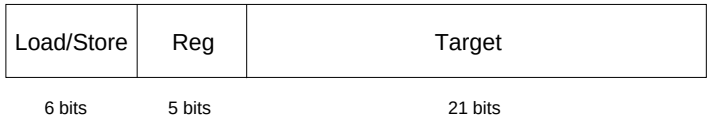
- Opcode for arithmetic instructions is 10_{16}
- Function code for addq instruction is 20_{16}
- Src1, Src2 and Dest each specify 1 of 32 registers

Example: bne



- Opcode for bne instruction is $3d_{16}$
- Test register specifies 1 of 32 registers for testing
- Only 21 bits left for target instruction address!!!

Load/Store Instructions



- How to specify memory address (*effective address*) with only 21 bits?