Assignment Project Exam Help Instruction Representation https://powcoder.com

Add WeChat powcoder

Review (1/2)

- Logical and Shift Instructions
 - Operate on bits individually, unlike arithmetic, which operate on entire word
 Use to isolate fields, either by masking or by shifting back and forth

 - Shift left logical (s11) multiplies by powers of m
 - Shift right arithmetic (sra) divides by powers of 2 close but strange roundagtoWregaratepownteder (e.g., $-5 \text{ sra } 2 \text{ bits} = -2 \text{ while } -5 / 2^2 = -5 / 4 = -1$)
- New Instructions: and andi or ori nor sll srl sra

Review (2/2)

- MIPS Signed versus Unsigned is an "overloaded" term
 - Do/Don't sign extend signment Project Exam Help (lb, lbu)
 https://powcoder.com
 - Don't overflow (addu, addiu, subu)
 Add WeChat powcoder
 - Compute the correct answer (multu, divu)
 - Do signed/unsigned compare (slt,slti/sltu,sltiu)

MULT vs MULTU

- In 2s complement, addition and subtraction are the same, as is the case in the low-half of a multiply.

 A full multiply, however, is not! Project Exam Help
- In 32-bit twos-complent the unsigned quantity 2^{32} 1. However:

 (-1)(-1) = +1

 (2^{32} 1)(2^{32} 1) = 2^{64} 2^{33} + 1

Overview

- Big idea: stored program

- MIPS instruction format for immediate, bata transfer instructions

Add WeChat powcoder

Big Idea: Stored-Program Concept

- Computers built on 2 key principles:
 - 1) Instructions are represented as numbers.
 - 2) Therefore stigrance project be stored in memory to be read or written just like numbers (data).
- Simplifies SW/HWPSf/cBmpcrder-systems:
 - Memory technology wedge data to be well for programs

Consequence #1: Everything Addressed

- Since all instructions and data are stored in memory as numbers, everything has a memory address
 - Instruction words and Saignment Project Exam Help
 - Both branches and jumps use these instruction addresses https://powcoder.com
- C pointers are just memory addresses: can point to anything in memory
 - Unconstrained use of addredses Wan Claat posses of addredses wan Claat posses of addredses.
 - Up to you in MIPS; up to you in C; limits in Java
- One register keeps address of instruction being executed:
 "Program Counter" (PC)
 - Just a pointer to memory: Intel calls it Instruction Address Pointer, a better name

Consequence #2: Binary Compatibility

- Programs are distributed in binary form
 - Programs bound to specific instruction set
 Assignment Project Exam Help
 Different version for Macintosh and BM PC
- New machines want tohtupso/powogramsoffbinaries") as well as programs compiled to new instructions Add WeChat powcoder
 • Leads to instruction set evolving over time
- Intel 8086 was selected in 1981 for 1st IBM PC
 - Latest PCs still use 80x86 instruction set...
 - Can (more or less) still run program from 1981 PC today!



Instructions as Numbers (1/2)

- All data we work with is in words (32-bit blocks):

 - Each register is a word.
 Assignment Project Exam. Help
 1w and sw both access memory one word at a time
- So how do we represent the strong der.com
 - Remember: Computer only understands 1s and 0s, so
 "add \$t0,\$0,\$0" is meaningless

 - MIPS wants simplicity:
 - Since data is in words, let the instructions be words too

Instructions as Numbers (2/2)

- Divide the 32-bit instruction word into "fields"
- Each field tells something about the instructional
- We could define different fields for each instruction, but MIPS is based on simplicity, so define Boasic types of instruction formats:
 - R-format

Add WeChat powcoder

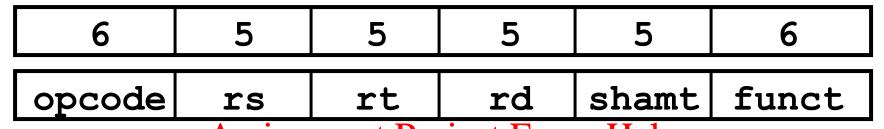
- I-format
- J-format (next lecture)

Instruction Formats

- I-format: used for instructions with immediates,

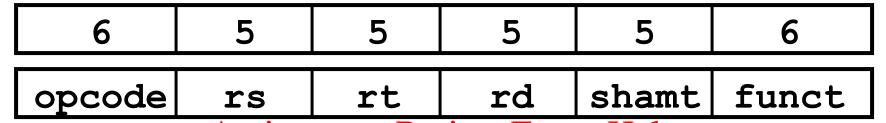
 - lw and sw (since the offset counts as an immediate),
 Assignment Project Exam Help
 beq and bne (branches use offsets as we will see later)
 - But not the shift instructions: (mpore entire leter)
- J-format: jump format used for j and jal Add WeChat powcoder
- R-format: used for all other instructions
 - R stands for register format
- It will soon become clear why the instructions have been partitioned in this way!

R-Format Instructions (1/5)



- Assignment Project Exam Help
 Break 32 bit "instruction" word into fields
 - For simplicith tapsh // plowhas denacrom
- Important: On these slides and in book, each field is viewed as a 5 or 6 bit unsigned integer, not as part of a 32 bit integer
 - 5 bit fields can represent any number 0-31,
 - 6 bit fields can represent any number 0-63.

R-Format Instructions (2/5)



• What do these field integer values tell us?

opcode partiblypspedples wide instruction it is

Note: This number is equal to 0 for all R-Format instructions. **funct** combined with **opcode**, this number exactly specifies the instruction

- Question:
 - Why aren't **opcode** and **funct** a single 12-bit field?
 - Think about it... We'll see the answer this later.



R-Format Instructions (3/5)



Assignment Project Exam Help

More fields

(Source Register). **generally** crs6010 specify register containing first operand

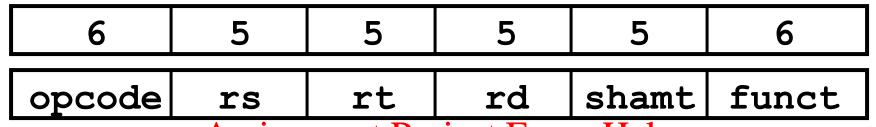
containing first operand

rt (Target Register): generally used to specify register

containing second operand (note that name is misleading)

<u>rd</u> (Destination Register): *generally* used to specify register which will receive **result of computation**

R-Format Instructions (4/5)

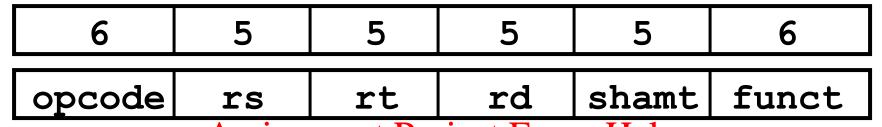


Assignment Project Exam Help
 Notes about register fields

- Each register fietoss/exactlycodets.com
- It can specify any unsigned integer in the range 0-31
 It specifies one of the 32 registers by number
- "generally" on previous slide because there are exceptions that we'll discuss more later...

mult and div have nothing important in the rd field since the dest registers are hi and lo mfhi and mflo have nothing important in the rs and rt fields since the source is determined by the instruction

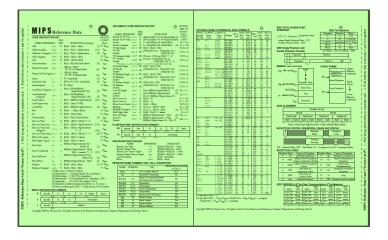
R-Format Instructions (5/5)



Assignment Project Exam Help
 One more field we have not yet discussed

Shamt containst power appount of shift instruction will shift Shifting a 32-bit word by more than 31 is useless, so this field is only 5 bits (selected) (

- This field is set to 0 in all but the shift instructions
- For a detailed description of field usage for each instruction, see green reference card in the textbook



R-Format Example (1/2)

• MIPS Instruction:

add \$8 \$9 \$10

Assignment Project Exam Help

opcode = 0 (look up in table)ttps://powcoder.e

funct = 32 (look up in table)
Add WeChat powco

rs = 9 (first operand)

rt = 10 (second operand)

rd = 8 (destination)

shamt = 0 (not a shift)



R-Format Example (2/2)

• MIPS Instruction:

```
$8 $9 $10
add
                                                                                                                                                           Assignment Project Exam Help
                               Decimal/field representation:
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         32
                               Binary/field representation Bi
                                       000000 01001 01010 01000 00000 100000
                                                                                                                                                                                                                                                                                                                                          012A4020<sub>hex</sub>
                                                                     hex representation:
                                                                                                                                                                                                                                                                                                                                          19546144<sub>ten</sub>
                                                                    decimal representation:
```

Called a Machine Language Instruction

I-Format Instructions (1/5)

- What about instructions with immediates?
 - 5-bit field only represents numbers up to the value 31
 Immediates may be much larger than this

 - Ideally, MIPS would have the instruction for simplicity)
 - Unfortunately, we need to compromise
- Define new instruction format partially consistent with R-format
 - Note that if the instruction has an immediate, then it uses at most 2 registers

I-Format Instructions (2/5)

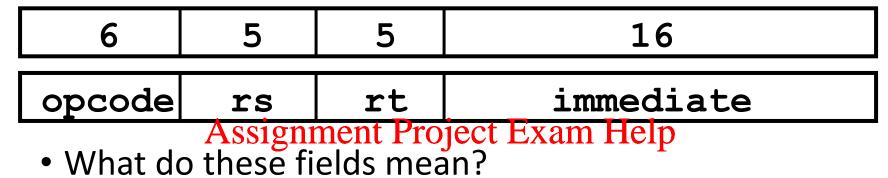


• Define "fields" of a fixed number of bits each 6 + 5 + 5 + 16 https://powcoder.com

• Again, each field that a color each field that a color each field that the color each field th

 Key Concept: Only one field is inconsistent with Rformat. Most importantly, opcode is still in same location.

I-Format Instructions (3/5)



- opcode: sahtelas: before, potewith the funct field, opcode uniquely specifies an instruction in I-format Add WeChat powcoder

 • This finally answers the question of why
- R-format has two 6-bit fields to identify instruction instead of a single 12-bit field...
 - In order to be consistent with other formats

I-Format Instructions (4/5)



More fields:

```
rs specifies the my register which will receive result of the computation (this is why it's called the target register "rt")
```

I-Format Instructions (5/5)



- The Immediate Field:
 - addi, sltihtspride Police is sign-extended to 32 bits. Thus, it's treated as a signed integer.
 16 bits → can be used to represent immediate up to 2¹⁶
 - different values
 - This is large enough to handle the offset in a typical 1w or sw, plus a vast majority of values that will be used in the **slti** instruction.

I-Format Example (1/2)

 MIPS Instruction: \$21 \$22 -50 Assignment Project addi https://powcod opcode = 8 (look up in table) Add WeChat p rs = 22(register containing operand) rt = 21(target register) immediate = -50(by default, specified in decimal)

		A				
	MIPS	Rο	for	ence Data	6	
				ence Data	4	
•	t de xvam	(N))		OPCODE
	NAME, MNEMO	NIC	FOR- MAT			/ FUNCT (Hex)
	Add	add	R	R[rd] = R[rs] + R[rt]	(I)	0 / 20 _{hex}
1	er com	addi	I	R[rt] = R[rs] + SignExtImm	(1,2)	8 _{hex}
J	Add Imm. Unsigned	addiu		R[rt] = R[rs] + SignExtImm	(2)	9 _{hex}
	Add Unsigned	addu	R	R[rd] = R[rs] + R[rt]	(-)	0 / 21 _{hex}
	And	and	R	R[rd] = R[rs] & R[rt]		0 / 24 _{hex}
1	owcode	r	I	R[rt] = R[rs] & ZeroExtImm	(3)	c _{hex}
Ĭ		2 41		if(R[rs]==R[rt])	(3)	
	Branch On Equal	beq	I	PC=PC+4+BranchAddr	(4)	4 _{hex}
	Branch On Not Equa	lbne	I	if(R[rs]!=R[rt]) PC=PC+4+BranchAddr	(4)	5 _{hex}
	Jump	j	J	PC=JumpAddr	(5)	2 _{hex}
	Jump And Link	jal	J	R[31]=PC+8;PC=JumpAddr	(5)	3 _{hex}
	Jump Register	jr	R	PC=R[rs]		0 / 08 _{hex}
	Load Byte Unsigned	lbu	I	R[rt]={24'b0,M[R[rs] +SignExtImm](7:0)}	(2)	24 _{hex}
	Load Halfword Unsigned	lhu	I	R[rt]={16'b0,M[R[rs] +SignExtImm](15:0)}	(2)	25 _{hex}
	Load Linked	11	I	R[rt] = M[R[rs] + SignExtImm]	(2,7)	30 _{hex}
	Load Upper Imm.	lui	I	R[rt] = {imm, 16'b0}	(-,-)	f _{hex}
	Load Word	lw	I	R[rt] = M[R[rs] + SignExtImm]	(2)	23 _{hex}
	Nor	nor	R	$R[rd] = \sim (R[rs] R[rt])$	(-)	0 / 27 _{hex}
	Or	or	R	R[rd] = R[rs] R[rt]		0 / 25 _{hex}
	Or Immediate	ori	I	R[rt] = R[rs] ZeroExtImm	(3)	d _{hex}

I-Format Example (2/2)

MIPS Instruction:

```
$21 $22 -50
Assignment Project Exam Help
Decimal/field representation:
```

```
8 22 21 -50
```

Binary/fieldhebresentattopp.wcoder

```
001000 10110 10101 1111111111001110
```

hexadecimal representation: 22D5FFCE_{hex}

decimal representation: 584449998_{ten}

I-Format Problem (1/3)

Problem:

- Chances are that adding he swands Lti will often use immediates small enough to fit in the immediate field
- But what if the value is https://powcoder.com
 - We need a way to Adea with tap 3 2 do the immediate in any I-format instruction!

I-Format Problems (2/3)

- Solution to Problem:

 - - Instead, add a new instruction to believe der.com
- New instruction:

register Add WeChat powcoder

- Stands for Load Upper Immediate
- Takes 16-bit immediate and puts these bits in the upper half (high order half) of the specified register
- Sets lower half word to zero

I-Format Problems (3/3)

Solution to Problem (continued):

- Now I-format instructions have only 16 bit immediates
- Assembler can do this for us automatically! (more on pseudoinstructions next lecture)

In conclusion

- Simplifying MIPS: Define instructions to be same size as data word (one word) so that they can use the same memory (compiler can use lw and sw).
 Assignment Project Exam Help
- Machine Language Instruction: 32 bits representing a single instruction

R	opcode	rs Add	WeCha	powcod rd	er shamt	funct
I	opcode	rs	rt	immediate		te

 Remember: The computer actually stores programs as a series of these 32-bit numbers.

Review and More Information

- TextBook
 - 2.5 Representing Instructions in the computer
 2.10 Addressing for 32-bit immediates

 - 2.12 Translating and Starting: 4/Brogrander.com
 - Just the section on the **Assembler** with respect to pseudoinstructions (pg 124, 125, 5th edition) Add WeChat powcoder

31 McGill COMP273