

Assignment Project Exam Help

Add WeChat powcoder


L3_3 LC-2K ISA

Assignment Project Exam Help

<https://powcoder.com>

EECS 370 – Introduction to Computer Organization – Fall 2020

Add WeChat powcoder



Assignment Project Exam Help

Learning Objectives

Add WeChat powcoder

- Recognize the set of instructions for LC-2K Architecture (ISA) and be able to describe the operations and operands for each instruction
- Ability to create simple LC-2K assembly programs, e.g., using addition and branching.

Assignment Project Exam Help

<https://powcoder.com>
- Understand and be able to replicate the encoding (translation from assembly to machine code) of instructions for any LC-2K assembly program

Add WeChat powcoder

Assignment Project Exam Help

LC-2K Processor

Add WeChat powcoder

always
contains
 \emptyset

Reg#
→ 0 ZR \emptyset
1
2
3
4
5
6
7

- 32-bit processor
 - Instructions are 32 bits
 - Integer registers are 32 bits
- 8 registers
- supports 65536 words of memory (addressable space)
- 8 instructions in the following common categories:
 - Arithmetic: **add**
 - Logical: **nor**
 - Data transfer: **lw, sw**
 - Conditional branch: **beq**
 - Unconditional branch (jump) and link: **jalr**
 - Other: **halt, noop**

Assignment Project Exam Help

<https://powcoder.com>

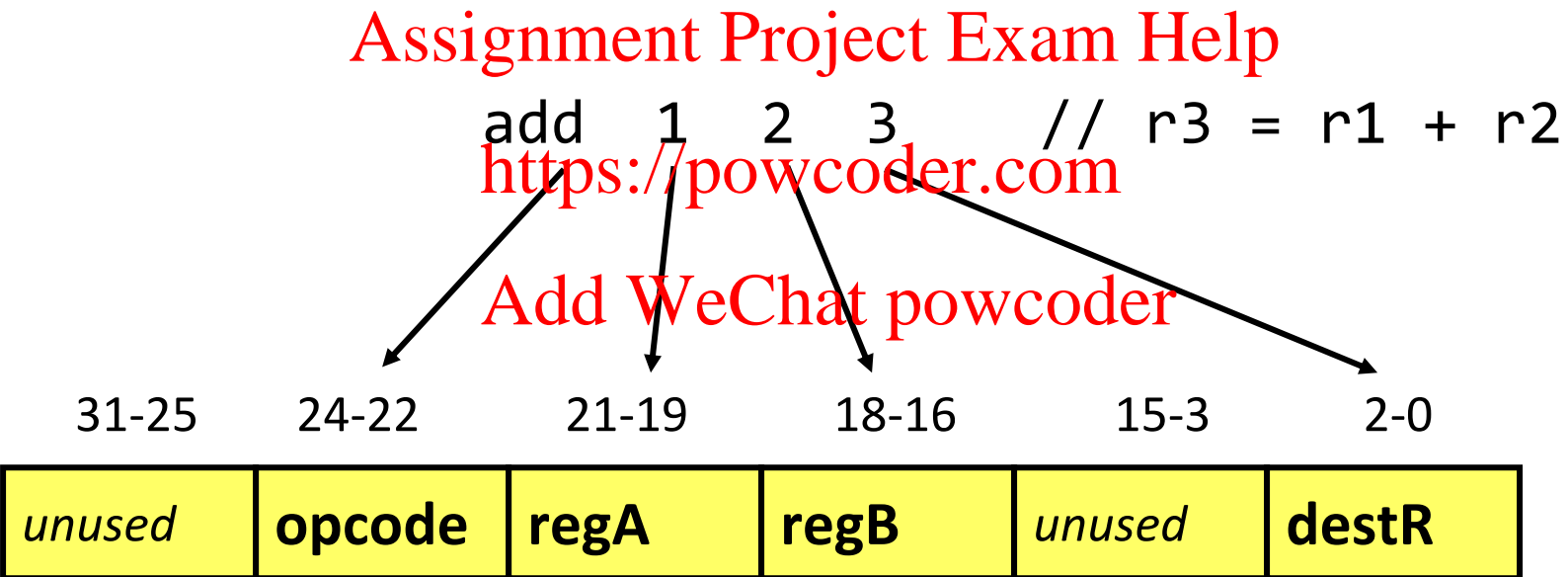
Add WeChat powcoder

Assignment Project Exam Help

Instruction Encoding

Add WeChat powcoder

- The Instruction Set Architecture (aka Architecture) defines the mapping of assembly instructions to machine code

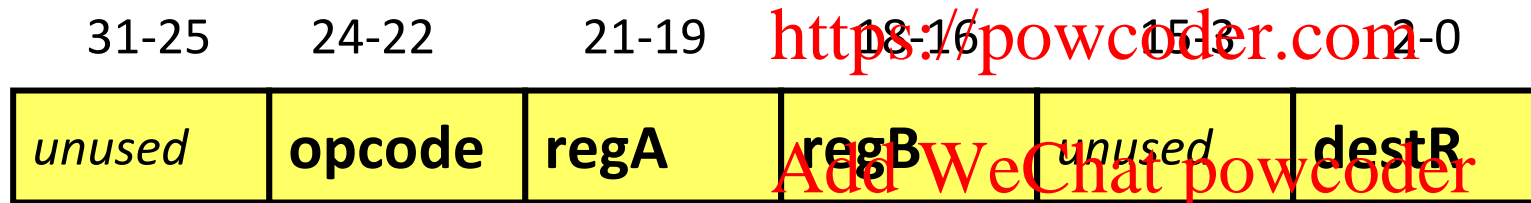


Assignment Project Exam Help

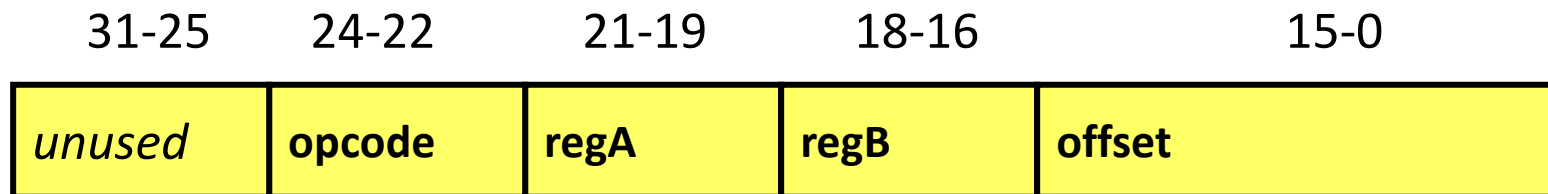
Instruction Formats – R-type, I-type

Add WeChat powcoder

- Tells you which bit fields correspond to which part of an assembly instruction
- R-type (register) – add (opcode 000), nor (opcode 001)



- I-type (immediate) - lw (opcode 010), sw (opcode 011), beq (opcode 100)

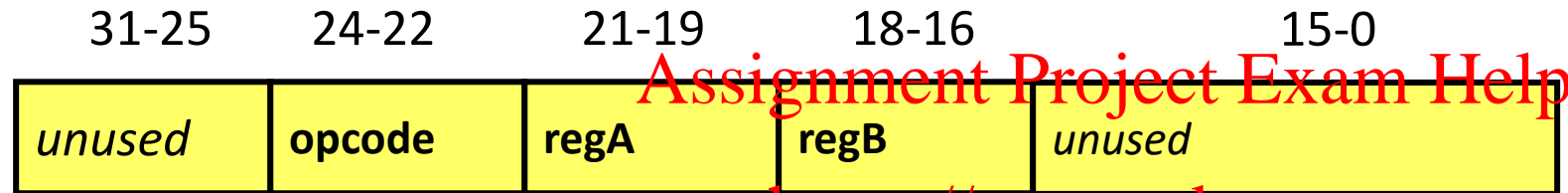


Assignment Project Exam Help

Instruction Formats – J-type, O-type

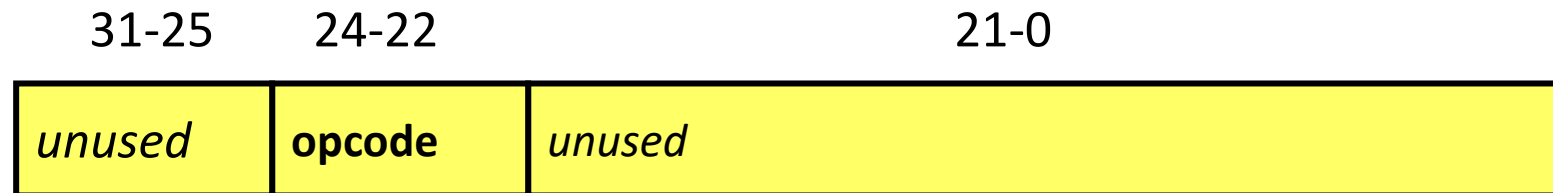
Add WeChat powcoder

- J-type (jump) – jalr (opcode 101)



<https://powcoder.com>

- O-type (???) - halt (opcode 110), noop (opcode 111)



Assignment Project Exam Help

Instruction Formats

Add WeChat powcoder

- The Instruction Set Architecture (aka Architecture) defines the mapping of assembly instructions to machine code

Instruction Type	Instruction	Bits 31-25	Bits 24-22	Bits 21-19	Bits 18-16	Bits 15-3	Bits 2-0
R-type	add	unused	opcode	reg A	reg B	unused	destReg
	nor						
I-type	lw					offsetField 16-bit, 2's complement number range:[-32768, 32767]	
	sw						
	beq						
J-type	jalr					unused	
O-type	halt						
	noop						

Unused: all unused bits should always be 0

Assignment Project Exam Help

Bit Encodings

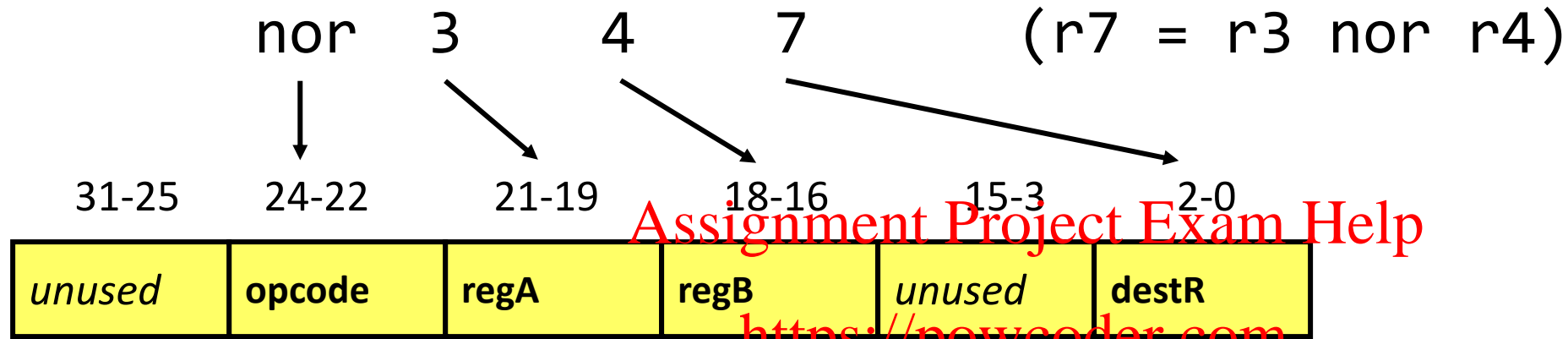
Add WeChat powcoder

- Opcode binary encodings:
 - add (000), nor (001), lw (010), sw (011), beq (100), jalr (101), halt (110), noon (111)
- Register operands
 - Binary encoding of register number, e.g., $r2 = 2 = 010$
- Immediate values
 - Binary encoding *using 2's complement values*
 - Give all available bits a value – *do not forget sign extension!*

Assignment Project Exam Help

Encoding Example #1 - nor

Add WeChat powcoder



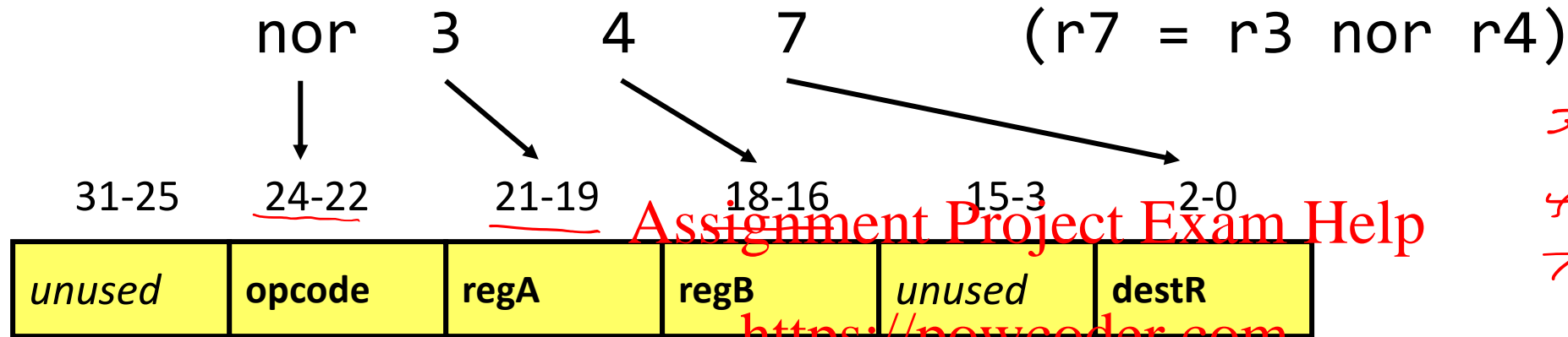
Add WeChat powcoder

Assignment Project Exam Help

Encoding Example #1 - nor

Add WeChat powcoder

opcode nr = 001



3 = 011

4 = 100

7 = 111

<https://powcoder.com>

0000_0000_0101100100000000000000000111

OP rA rB destR

hex 0 5 C 0 0 7

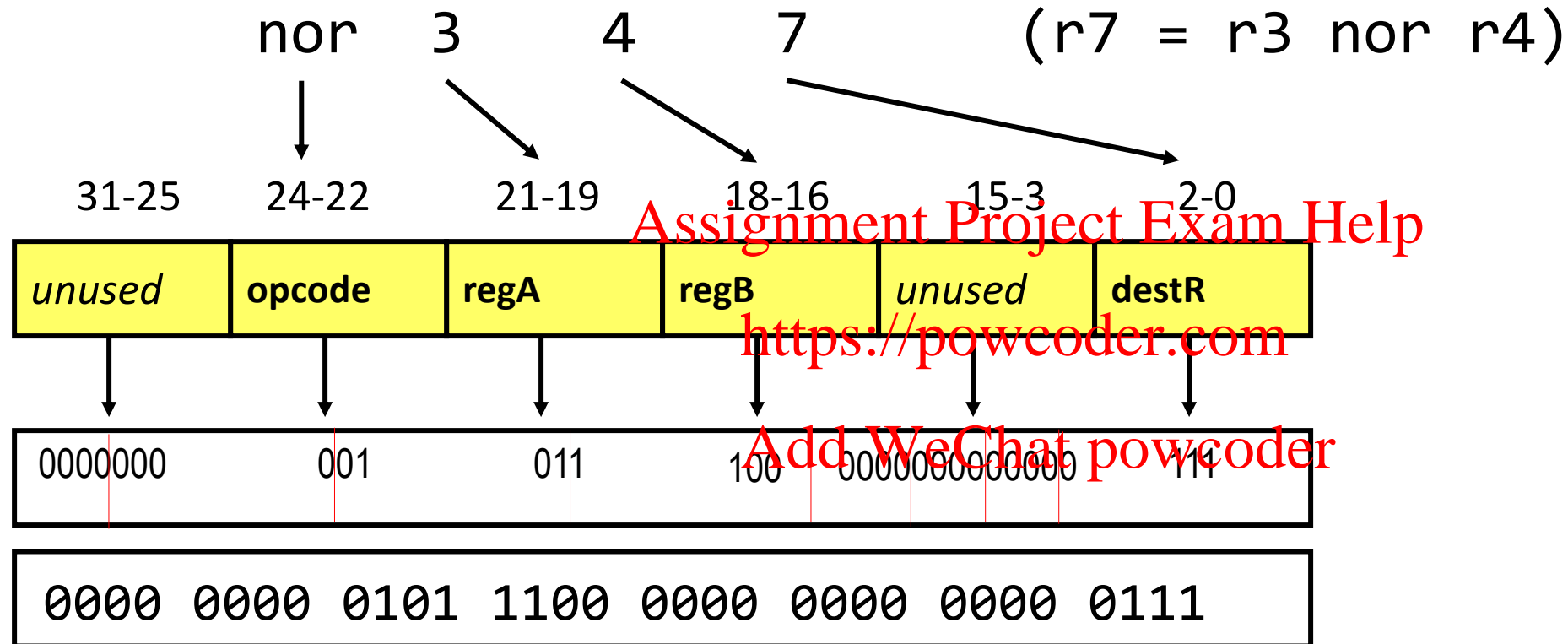
$$2^{22} + 2^{20} + 2^{19} + 2^{18} + 2^2 + 2^1 + 2^0$$

6029319

Assignment Project Exam Help

Encoding Example #1 - nor

Add WeChat powcoder



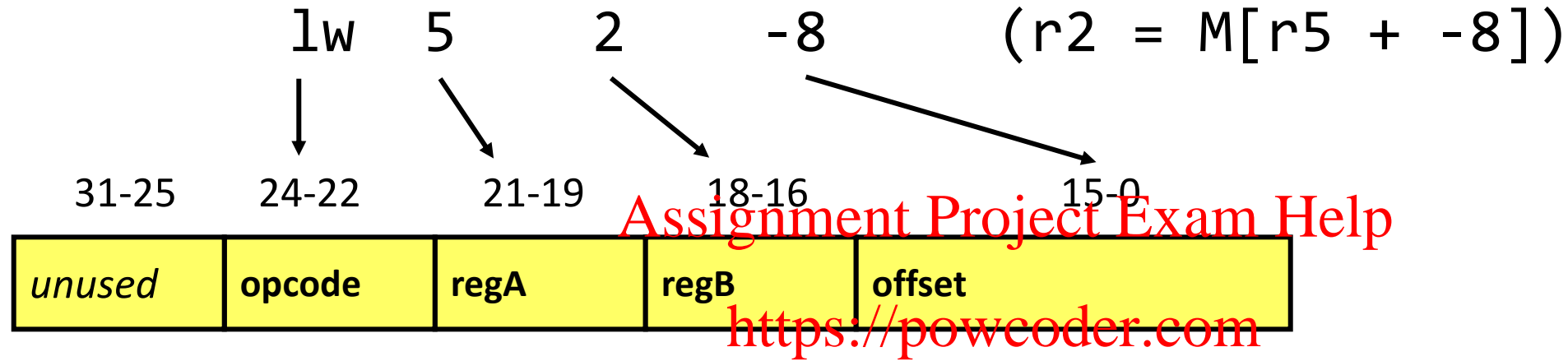
Convert to Hex → 0x005C0007

Convert to Dec → 6029319

Assignment Project Exam Help

Encoding Example #2 - lw

Add WeChat powcoder

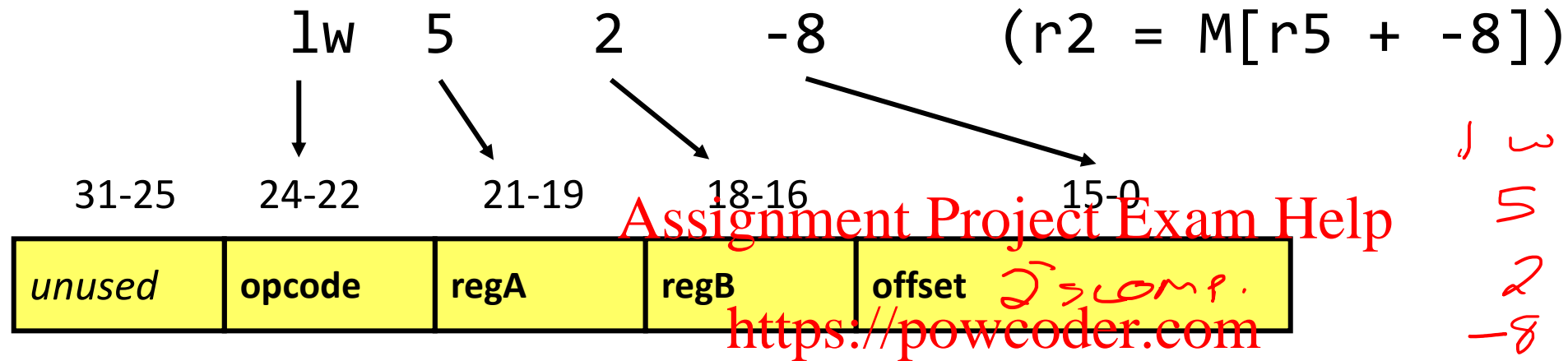


Add WeChat powcoder

Assignment Project Exam Help

Encoding Example #2 - lw

Add WeChat powcoder



1w = 010
5 = 101
2 = 010
-8 = ???

Add WeChat powcoder

0000 0000 0101 0101 1111 1111 1111 1000

0C rA rB offset

-8

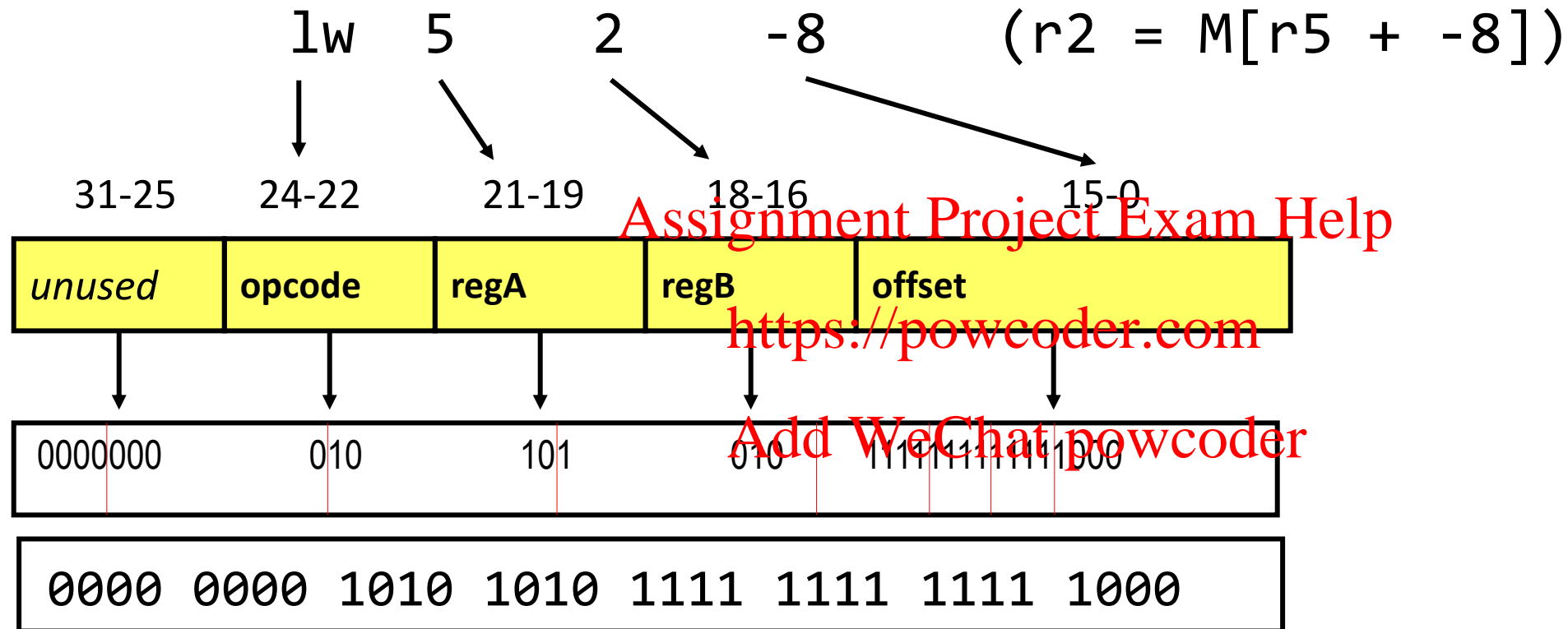
1111 1111 1111 1000

hex 0 4 A A F F F 8 →

Assignment Project Exam Help


Encoding Example #2 - lw

Add WeChat powcoder



Convert to Hex → 0x00AAFF8

Convert to Dec → 11206648



Assignment Project Exam Help

Encoding Example #3 - add

Add WeChat powcoder


- Compute the encoding in Hex for:

add 3 7 3 (r3 = r3 + r7) (add = 000)

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



Assignment Project Exam Help

Encoding Example #3 - add

Add WeChat powcoder

- Compute the encoding in Hex for:
add 3 7 3 (r3 = r3 + r7) (add = 000)

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Assignment Project Exam Help
Encoding Example #3 - add
Add WeChat powcoder

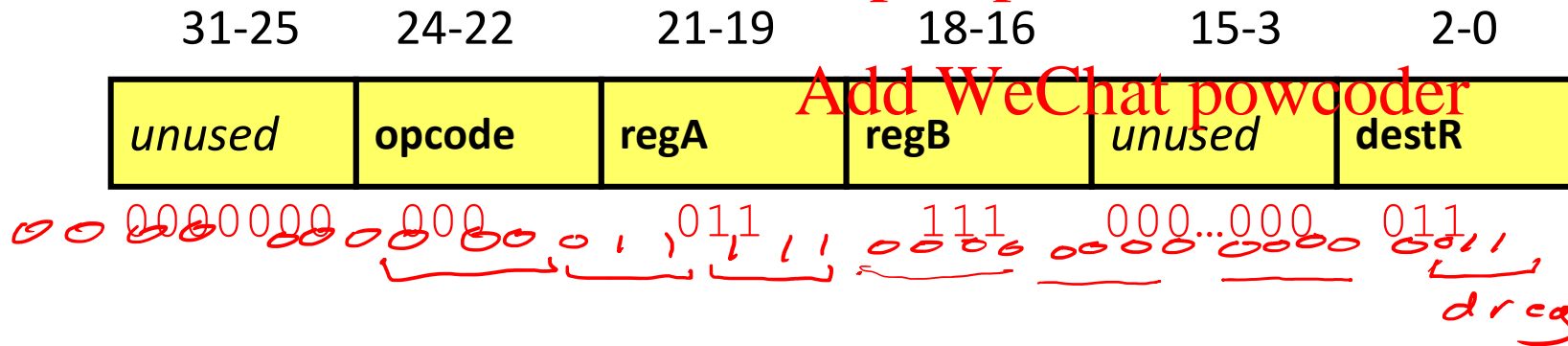
add	000
3	011
7	111

- Compute the encoding in Hex for:

```
add    3    7    3    (r3 = r3 + r7)    (add = 000)
```


Assignment Project Exam Help

<https://powcoder.com>



Convert to Hex → 0x001F0003

Convert to Dec → 2031619



Assignment Project Exam Help

Encoding Example #4 - SW

Add WeChat powcoder

- Compute the encoding in Hex for:

sw 1 5 67 (M[r1+67] = r5) (sw = 011)

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Assignment Project Exam Help

Encoding Example #4 - SW

Add WeChat powcoder

- Compute the encoding in Hex for:

sw 1 5 67 (M[r1+67] = r5) (sw = 011)

sw = 011

1 = 001

5 = 101

67

Assignment Project Exam Help

<https://powcoder.com>

64 + 2 + 1

00000000 01000000 00000000 00000001

Assignment Project Exam Help

Encoding Example #4 - SW

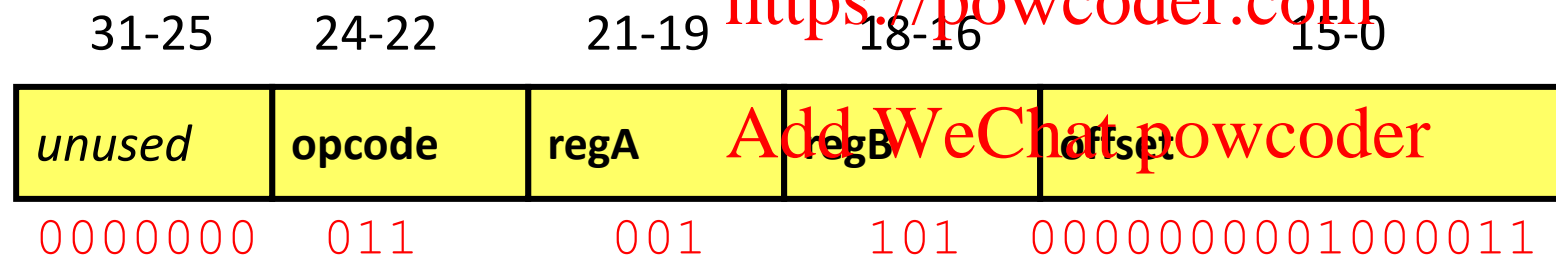
Add WeChat powcoder

- Compute the encoding in Hex for:

sw 1 5 67 (M[r1+67] = r5) (sw = 011)

Assignment Project Exam Help

<https://powcoder.com>



Convert to Hex → 0x00CD0043

Convert to Dec → 13434947



Assignment Project Exam Help

Assembler, aka, P1a

Add WeChat powcoder

- Each line of assembly code corresponds to a number
 - “add 0 0 0” is just 0.
 - “lw 5 2 -8” is 11206648
- Assembly code is how people write instructions for an ISA
 - We only use assembly because it's easier to read.
- Assembly code must be assembled (instructions encoded) to machine code for execution

Assignment Project Exam Help

Assembler Directive - `.fill`

Add WeChat powcoder

- You might want a number to be, well, a number.
 - Data for `lw`, `sw` instructions will be added to LC-2K assembly code file
- `.fill` tells the assembler to put a number instead of an instruction
- The syntax (to have a value of 7) is just `.fill 7`
- Question:
 - What do `.fill 7` and `add 0 0 7` have in common?


Assignment Project Exam Help

Assembler Directive - `.fill`

Add WeChat powcoder

- You might want a number to be, well, a number.
 - Data for `lw`, `sw` instructions will be added to LC-2K assembly code file
- `.fill` tells the assembler to put a number instead of an instruction
- The syntax (to have a value of 7) is just `.fill 7`
- Question:
 - What do `.fill 7` and `add 0 0 7` have in common?

They have the same value in machine code: 7 (decimal) 111 (binary)
really 0000 0000 0000 0000 0000 0000 0000 0111



Assignment Project Exam Help

Labels in LC-2K

Add WeChat powcoder

- Labels are used in lw/sw instructions or beq instruction

- For lw or sw instructions, the assembler should compute offsetField to be equal to the address of the label
 - i.e. $\text{offsetField} = \text{address of the label}$
- For beq instructions, the assembler should translate the label into the numeric offsetField needed to branch to that label
 - i.e. $\text{PC} + 1 + \text{offsetField} = \text{address of the label}$

Assignment Project Exam Help

Labels in LC-2K – Example #1

Add WeChat powcoder

- Labels are a way of referring to a line in an assembly-language program

0 loop beq 3 4 end

1 noop

2 ~~beq~~ 1 3 loop

3 end halt

Handwritten notes and calculations:

$loop: 0$
 $end: 3$

$beq: 100$
 011
 $4 \quad 100$

$beq \ 3 \ 4 \ 3$

$off + PC + 1 = 3$
 $off + 0 + 1 = 3$
 $off + 1 = 3$
 $off = 2$

$PC + 1 + off = 0$
 $2 + 1 + off = 0 \quad off = -3$

Diagram of LC-2K instruction format:

$\underbrace{100}_{OP} \underbrace{00}_{rA} \underbrace{1011}_{rB} \underbrace{\quad}_{off}$

Assignment Project Exam Help

Labels in LC-2K – Example #1

Add WeChat powcoder

- Labels are a way of referring to a line in an assembly-language program

```
loop    beq    3    4    end
        noop
        beq    1    3    loop
end      halt
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Assignment Project Exam Help

Labels in LC-2K – Example #1

Add WeChat powcoder

- Labels are a way of referring to a line in an assembly-language program

```
loop  beq  3  4  end
      noop
      beq  1  3  loop
end    halt
```

Assignment Project Exam Help

Replacing use of labels with values
<https://powcoder.com>

Add WeChat powcoder

loop is address 0
end is address 3

1st Pass

Addresses Instructions

0	loop	beq	3	4	2
1		noop			
2		beq	1	3	-3
3	end	halt			

2nd Pass

Assignment Project Exam Help

Program in LC-2K – Example

Add WeChat powcoder

1. Encode program instructions 2. What does this program do?

```
loop  lw    0    1    one
      add   1    1    1
      sw    0    1    one
      halt
one   .fill  1
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Assignment Project Exam Help

Program in LC-2K – Example

Add WeChat powcoder

1. Encode program instructions
2. What does this program do?

loop lw 0 1 ~~one~~ 4
add 1 1 1
sw 0 1 ~~one~~
halt
one .fill 1

1st Pass
one : 4
<https://powcoder.com>

Add WeChat powcoder

Assignment Project Exam Help

Program in LC-2K – Example

Add WeChat powcoder

1. Encode program instructions
2. What does this program do?

test.as

test.mc

loop	lw	0	1	one	8454148
	add	1	1	1	589825
	sw	0	1	one	12648452
	halt				25165824
one	.fill	1			1

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



Logistics

Assignment Project Exam Help
Add WeChat powcoder

- This is the final of 3 videos for lecture 3
 - L3_1 – ISAs – Instructions and Memory
 - L2_2 – Two's Complement
 - L2_3 – LC-2K ISA
- There are two worksheets for lecture 3
 1. Addressing and 2's complement
 2. LC-2K program encoding
- Complete the participation quiz for lecture 3 on Canvas
 - Due by 9/13 at 11:59 pm