MIPS Instructions, Memory Storage

General Purpose Registers

REGISTER NAME	REGISTER #	DESCRIPTION
\$zero	\$0	a special purpose register which always contains a constant value of 0. It can be read, but cannot be written.
\$at	\$1	a register reserved for the assembler. If the assembler needs to use a temporary register (e.g. for pseudo instructions), it will use \$at, so this register is not available for use programmer use.
\$v0-\$v1	\$2-\$3	registers are normally used for return values for subprograms. \$v0 is also used to input the requested service to syscall.
\$a0-\$a3	\$4-\$7	registers are used to pass arguments (or parameters) into subprograms.
\$t0-\$t9	\$8-\$15, \$24-\$25	registers are used to store temporary variables. The values of temporary variables can change when a subprogram is called.
\$s0-\$s7	\$16-\$23	registers are used to store saved values. The values of these registers are maintained across subprogram calls.
\$k0-\$k1	\$26-\$27	registers are used by the operating system, and are not available for use programmer use.
\$gp	\$28	pointer to global memory. Used with heap allocations.
\$sp	\$29	stack pointer, used to keep track of the beginning of the data for this method in the stack.
\$fp	\$30	frame pointer, used with the \$sp for maintaining information about the stack. This text will not use the \$fp for method calls.
\$ra	\$31	return address, a pointer to the address of the instruction to execute when returning from a subprogram.

Source: Introduction To MIPS Assembly Language Programming by Charles W. Kann, 2015

Special	Registers
рс	
hi	After a mult instruction -
	After a div instruction
10	After a mult instruction -
	After a div instruction
Note You may	use either the name or register number when writing a program.
Example	2
Assemb	eler Directives
	ed by a
•	

^{© 2019,} Rebecca Rashkin - This document may be copied, redistributed, transformed, or built upon in any format for **educational**, non-commercial purposes. Please give me appropriate credit should you choose to modify this resource. Thank you:)

Data Directives

Description	Stored As					
	ADDRESS	CONTENTS				
	label + 4					
	label + 3					
General Format	label + 2					
	label + 1					
	label + 0					
Example Usage						

.ascii

Description	Stored As				
	ADDRESS	CONTENTS			
General Format	label + 2				
defici al Tormac	label + 1				
	label + 0				
Example Usage					

[©] 2019, Rebecca Rashkin - This document may be copied, redistributed, transformed, or built upon in any format for **educational**, non-commercial purposes. Please give me appropriate credit should you choose to modify this resource. Thank you:)

Description	Stored As	
	ADDRESS	CONTENTS
	label + 4	
General Format	label + 3	
	label + 2	
	label + 1	
	label + 0	
Example Usage		

.bvte

Description	Stored As				
	ADDRESS CO	ONTENTS			
General Format	label + 5				
	label + 4				
General Format	label + 3				
	label + 2				
	label + 1				
	label + 0				
Example Usage					

^{© 2019,} Rebecca Rashkin - This document may be copied, redistributed, transformed, or built upon in any format for **educational**, non-commercial purposes. Please give me appropriate credit should you choose to modify this resource. Thank you:)

half	
Description	Stored As
	ADDRESS CONTENTS
	label + 5
	label + 4
General Format	label + 3
	label + 2
	label + 1
	label + 0
Example Usage	
.word	
Description	Stored As
	ADDRESS CONTENTS
	label + 7

General Format

ADDRESS CONTENTS

label + 7

label + 6

label + 5

label + 4

label + 3

label + 2

label + 1

label + 0

Example Usage

^{© 2019,} Rebecca Rashkin - This document may be copied, redistributed, transformed, or built upon in any format for **educational**, non-commercial purposes. Please give me appropriate credit should you choose to modify this resource. Thank you:)

.float

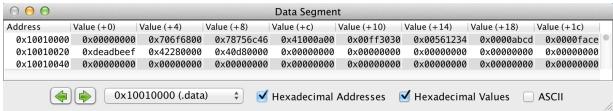
Description	Stored As						
	ADDRESS	CONTENTS					
	label + 7						
	label + 6						
General Format	label + 5						
	label + 4						
	label + 3						
	label + 2						
	label + 1						
	label + 0						
Example Usage							
42: 0x42280000 6.75: 0x40D80000							

Example

(an expanded version of this example can be found here)

.data

```
.space 5
.ascii "hop"
.asciiz "Flux"
.byte 10 0x00 0x41 48 0x30 0xFF
.half 0x1234 0x56 0xABCD
.word 0xFACE 0xDEADBEEF
.float 42 6.75
```



^{© 2019,} Rebecca Rashkin - This document may be copied, redistributed, transformed, or built upon in any format for **educational**, non-commercial purposes. Please give me appropriate credit should you choose to modify this resource. Thank you:)

Conver	ted by	the	assemb	oler to an					 			
Indica	ated by	a										
Used t	o label	L					an	d				
Can be	e used i	in _				and _			 	_segment	s of	code.
Samp	le Prog	ram										
.text												
	addiu	\$t0	\$zero	0	#	instr 1						
	li	\$t1	2		#	instr 2						
loop:	nop				#	instr 3						
	addiu	\$t0	\$t0	1	#	instr 4						
	bne	\$t0	\$t1	loop	#	instr 5						
	nop				#	instr 6						

				Text S	egm	nent							
Program Arguments:													
Bkpt	Address	Code	Basic		Sou	ırce							
	0x00400000	0x24080000	addiu	\$8,\$0,0x00000000	2:		addiu	\$t0	\$zero	0	#	instr 1	
	0x00400004	0x24090002	addiu	\$9,\$0,0x00000002	3:		li	\$t1	2		#	instr 2	
	0x00400008	0×00000000	nop		5:	loop:	nop				#	instr 3	}
	0x0040000c	0x25080001	addiu	\$8,\$8,0x00000001	6:		addiu	\$t0	\$t0	1	#	instr 4	ļ
	0x00400010	0x1509fffd	bne \$8	3,\$9,0xfffffffd	7:		bne	\$t0	\$t1	loop	#	instr 5	,
	0x00400014	0×00000000	nop		9:		nop				#	instr 6	j

instruction address - machine code - native commands - written by programmer

^{© 2019,} Rebecca Rashkin - This document may be copied, redistributed, transformed, or built upon in any format for **educational**, non-commercial purposes. Please give me appropriate credit should you choose to modify this resource. Thank you:)

MIPS Address Spac	ce (Not to Scale)
0x9000 0000	MMIO
	Kernel Data
0.7555 5-00	
0x7fff fe00	Stack
	Неар
0x1004 0000	
0x1001 0000	Static Data
	Program Text
0x0040 0000	
0×0000 0000	Reserved
The label loop rep	resents which address?
Instruction Memo	ry
MIPS ISA is	

A program is comprised of _____

choose to modify this resource. Thank you:)

^{© 2019,} Rebecca Rashkin - This document may be copied, redistributed, transformed, or built upon in any format for **educational**, non-commercial purposes. Please give me appropriate credit should you

which are encoded as __			_(;
Therefore, each instru Example		mem	ory lo	catio	ons.			
0x00400004 0x24090 0x00400008 0x000000 0x0040000c 0x25080	001 addiu \$8,\$8,0x00000001 ffd bne \$8,\$9,0xfffffffd	3: 5: loop:	: nop	\$t1 \$t0	2 \$t0		<pre># instr # instr # instr # instr # instr # instr</pre>	2 3 4 5
What is stored at eac addresses? ADDRESS 0x0040 0014 0x0040 0010 0x0040 000C 0x0040 0008 0x0040 0004 0x0040 0000			ser loc 00000 t SSS 0007 0006 0005 0004 0003 0002	_		x004000		[R)

^{© 2019,} Rebecca Rashkin - This document may be copied, redistributed, transformed, or built upon in any format for **educational**, non-commercial purposes. Please give me appropriate credit should you choose to modify this resource. Thank you:)

Instruction Format					
Many instructions use registers: 1	& 2				
Write the general form of an instruction:					
Write an example of an instruction:					
Instruction Types					
1					
2 -					
3					
- 					
Operate Instructions					
Uses	unless unsigned is indicated.				
e.g					
Example: Assume \$t0 = 0xFFFFFFFF, \$t1 = 0x00000001					
add \$t2 \$t0 \$t1	addu \$t2 \$t0 \$t1				
To those eventless	To those eventless				
Is there overflow?	Is there overflow?				

^{© 2019,} Rebecca Rashkin - This document may be copied, redistributed, transformed, or built upon in any format for **educational**, non-commercial purposes. Please give me appropriate credit should you choose to modify this resource. Thank you:)

Register vs. Immediate Instructions

e.g.

add \$t2 \$t1 \$t0	addi \$t2 \$t1 0x4

In MIPS, the immediate value can be _____ bits max.

Other Operate Instructions

e.g. _____

Loads and Stores

Load Word

Loads	of data
from memory into a	
What is the value of \$t0? LI \$t1 0x1010 LW \$t0 (\$t1)	
\$t0 = 0x	

ADDRESS	CONTENTS
0x1013	0xC0
0x1012	0xFF
0x1011	0xEE
0x1010	0xEE

Store Word

Takes a _____ and stores the value in _____

Example

What does memory look like after these instructions?

LI \$t1 0x1010 LI \$t0 0xABCDEF00 SW \$t0 (\$t1)

ADDRESS	CONTENTS			
	BEFORE	AFTER		
0x1013	0xC0			
0x1012	0xFF			
0x1011	0xEE			
0x1010	0xEE			

^{© 2019,} Rebecca Rashkin - This document may be copied, redistributed, transformed, or built upon in any format for **educational**, non-commercial purposes. Please give me appropriate credit should you choose to modify this resource. Thank you:)

CONTENTS

0xFE

0xED

0xBA

0xBE

Load Half Word

Loads a()	ADDRESS
from memory into a	0x1013
Uses	0x1012
Example What is the value of \$t0?	0x1011
LI \$t1 0x1010 LH \$t0 (\$t1)	0x1010
\$t0 =	
\$t0 = 0x	

Store Half Word

Stores the
()
from a register into
Example What does memory look like after these instructions?
LI \$t1 0x1010 ADDI \$t0 \$zero 0xEF00 SH \$t0 (\$t1)
Land Date

ADDRESS	CONTENTS			
	BEFORE	AFTER		
0x1013	0xFE			
0x1012	0xED			
0x1011	0xBA			
0x1010	0xBE			

Load Byte

Loads a
from memory into a
Uses
Example What is the value of \$t0? LI \$t1 0x1010 LB \$t0 (\$t1)
\$t0 =
\$t0 = 0x

ADDRESS	CONTENTS
0x1013	0xFE
0x1012	0xED
0x1011	0xBA
0x1010	0xBE

^{© 2019,} Rebecca Rashkin - This document may be copied, redistributed, transformed, or built upon in any format for **educational**, non-commercial purposes. Please give me appropriate credit should you choose to modify this resource. Thank you:)

S	+	^	r	0		R	١.	,+	-
_	L	U		C	- 1	ט	v	Ľ	C

ADDRESS	CONTENTS	CONTENTS		
	BEFORE	AFTER		
0x1013	0xFE			
0x1012	0xED			
0x1011	0xBA			
0x1010	0xBE			
	0x1013 0x1012 0x1011	BEFORE 0x1013 0xFE 0x1012 0xED 0x1011 0xBA		

[©] 2019, Rebecca Rashkin - This document may be copied, redistributed, transformed, or built upon in any format for **educational**, non-commercial purposes. Please give me appropriate credit should you choose to modify this resource. Thank you:)

Example

				Text S	Segmer	nt							
Prog	ram Argument	is:											
Bkpt	Address	Code	Basic		Sourc	:e							
	0x00400000	0x24080000	addiu	\$8,\$0,0x00000000	2:		addiu	\$t0	\$zero	0	#	instr	1
	0×00400004	0x24090002	addiu	\$9,\$0,0x0000000	2 3:		li	\$t1	2		#	instr	2
	0x00400008	0x00000000	nop		5: lc	oop:	nop				#	instr	3
	0x0040000c	0x25080001	addiu	\$8,\$8,0x0000000	6:		addiu	\$t0	\$t0	1	#	instr	4
	0x00400010	0x1509fffd	bne \$8	3,\$9,0xfffffffd	7:		bne	\$t0	\$t1	loop	#	instr	5
	0x00400014	0×00000000	nop		9:		nop				#	instr	6

Instruction Execution Order

Assume each instruction takes 1 clock cycle to execute.

CLOCK CYCLE	PC	INSTRUCTION
1		
2		
3		
4		
5		
6		
7		
8		
9		

Example

Assembly code

^{© 2019,} Rebecca Rashkin - This document may be copied, redistributed, transformed, or built upon in any format for **educational**, non-commercial purposes. Please give me appropriate credit should you choose to modify this resource. Thank you:)

Endian-ness

The storage order of bytes in memory is determined by the _____

Example

LI \$t1 0x1014		How should we store	the bytes in memory?
LI \$t0 0x12345678 SW \$t0 (\$t1)	ADDRESS	CONTENTS	
	0x1017		
	0x1016		
		0x1015	
		0x1014	

Little Endian

	stored at the	ADDRESS	CONTENTS
	memory address	0x1017	
Note		0x1016	
The order of		0x1015	
	within each	0x1014	
	stays the		
same.			
Example			
LI \$t1 0x1014			
LI \$t0 0xABCDEF00			
SW \$t0 (\$t1)			
\$t0 = 0x			

^{© 2019,} Rebecca Rashkin - This document may be copied, redistributed, transformed, or built upon in any format for **educational**, non-commercial purposes. Please give me appropriate credit should you choose to modify this resource. Thank you:)

	na	

stored at the	A
memory address	0:
Note	
The order of	0:
within each	0:
stays the same.	0:
Example	
LI \$t1 0x1014 LI \$t0 0xABCDEF00 SW \$t0 (\$t1)	

ADDRESS	CONTENTS
0x1017	
0x1016	
0x1015	
0x1014	

Examples

Assuming LI \$t1 LW \$t0	
\$t0 = 0x	
Assuming LI \$t1 LW \$t0	
\$t0 = 0x	

ADDRESS	CONTENTS
0x1015	0x34
0x1014	0x12
0x1013	0xFE
0x1012	0xED
0x1011	0хBA
0x1010	0xBE

Note

MIPS uses _____ memory storage format.

What happens if we try to execute:
LI \$t1 0x1011
LW \$t0 (\$t1)

© 2019, Rebecca Rashkin - This document may be copied, redistributed, transformed, or built upon in

^{© 2019,} Rebecca Rashkin - This document may be copied, redistributed, transformed, or built upon in any format for **educational**, non-commercial purposes. Please give me appropriate credit should you choose to modify this resource. Thank you:)

Alignment

ADDRESS	CONTENTS	←		
0x101C	0x11			
0x101B	0xEF	←		
0x101A	0xCD			
0x1019	0xAB	←MIPS permits only		
0x1018	0x90			
0x1017	0x78			
0x1016	0x56			
0x1015	0x34	memory accesses for		
0x1014	0x12			
How do we	determine if an access is memor	ry aligned?		
For word a	alignment, look at			
For half v	word alignment, look at			
Which of th	o following momory accossos are align	od?		
	e following memory accesses are align			
LW \$t0 (6		SH \$t0 (0x1011)		
LW \$t0 (6	·	SH \$t0 (0x1019)		
SW \$t0 (6	·	LB \$t0 (0x1011)		
·	9x1022)	LBU \$t0 (0x1014)		
LH \$t0 (0	9x1011)	SB \$t0 (0x1015)		
LHU \$t0 (0	9x1013)	SB \$t0 (0x1017)		
What happe	ens if we try to make an unaligr	ned memory access?		

[©] 2019, Rebecca Rashkin - This document may be copied, redistributed, transformed, or built upon in any format for **educational**, non-commercial purposes. Please give me appropriate credit should you choose to modify this resource. Thank you:)

e.g	Pseudo Instructions (aka)							
Some instructions have several formats. e.g								
e.g	e.g							
Instruction Encoding MIPS instructions are encoded in bits. There are different instruction formats in MIPS. How many general purpose registers does MIPS have? So, how many bits do we need for the registers? type (Some instructions have sever	al formats.						
MIPS instructions are encoded in bits. There are different instruction formats in MIPS. How many general purpose registers does MIPS have? So, how many bits do we need for the registers? type (type) type (type) e.g	e.g							
There are different instruction formats in MIPS. How many general purpose registers does MIPS have? So, how many bits do we need for the registers? type (type) type (type) e.g	Instruction Encoding							
How many general purpose registers does MIPS have? So, how many bits do we need for the registers?	MIPS instructions are encode	d in bit	ts.					
So, how many bits do we need for the registers?	There are different instruction formats in MIPS.							
	How many general purpose reg	isters does MI	IPS have?					
e.g	So, how many bits do we need	for the regis	sters? _					
e.g	type (_ type)					
type (type)e.g								
type (type)e.g								
e.g	e.g							
e.g								
	type (1	_ type) 		I			
type (type)	e.g							
	type (type)					
			_ 31- /					
e.g	e.g							

^{© 2019,} Rebecca Rashkin - This document may be copied, redistributed, transformed, or built upon in any format for **educational**, non-commercial purposes. Please give me appropriate credit should you choose to modify this resource. Thank you :)

Example: What types of instruct	ions are the following i	nstructions?	
ADD	LI	BLEZ	
BEQ	SH	J	
ADDI	LB	JR	
How do pseudo ops get encoded	as machine code?		
Examples			
Decode this instruction: 0	0x00AF8020		
Decode this instruction: (
Decode this instruction.	3X2 1020031		
Decode this instruction: (2,40000000		
becode this instruction.	220000000		

^{© 2019,} Rebecca Rashkin - This document may be copied, redistributed, transformed, or built upon in any format for **educational**, non-commercial purposes. Please give me appropriate credit should you choose to modify this resource. Thank you:)