ENGG 3380: Computer Organization and Design

Project Phase Two

Group	p #: 6
Name	Student ID
Elijah Johnston	0961152
Thomas Green	1048389
Lorent Aliu	1058897

Contents

Overview	3
ALU Configuration	
-	
Control Unit Configuration	4
Changes from Phase 1	5
Appendix	7

Overview

To begin, before designing the control path it was necessary to design the overall microprocessor, as this higher-level design dictates what control signals exists and hence the contents/layout of the microcode store; Figure 1 shows the block diagram of such. Evidently, this design is quite similar to that of the MIPS 32 architecture. There are a few changes, however. To begin, there exists "special register file" and "B selector", as will be further discussed under "Changes from Phase 1" section. Furthermore, there is no need for an ALU control unit, as all its function can be implemented using the microcode store. Finally, there is no logic needed for the computation of the jump address, as the 12 bit "target" field represents an address directly. This design uses 8-bit registers and 16-bit memory, as outlined in phase 1.

ALU Configuration

High Level Design

The ALU designed for the purpose of this project is very similar to that created for lab 4. To begin, it is split into the following blocks: An arithmetic unit, a logic unit, a conditional unit, and a "master mux" (figure 2). The three main units compute their respective results on any given clock cycle, while the master mux chooses which one to output provided what the select lines are set to.

The Arithmetic Unit

The arithmetic unit uses cascading 1-bit units to perform all its main operations (figure 3). For addition, the unit operates as a typical ripple carry adder would. For subtraction operations, operand B is complemented (as it is XOR'd with SO = 1), and SO acts as a carry in. The overflow detector is much simpler than the one implemented for lab 4. Carry 7 and 8 are XOR'd together to detect whether there is overflow for signed operations (this means that the result's sign is different from that of the operands, which have identical signs). For unsigned operations, we are simply interested in whether the final carry out is 1. Finally, status signal zero (used in BEQ) is implemented by NORing all of the result bits.

The Logic Unit

The logic unit takes advantage of eight 1-bit logic units to perform its operations (figure 6). Each 1-bit unit is simply a 3-1 mux (figure 7). The inputs of the mux represent each operation (ADD, OR and XOR). The select lines S1 and S0 choose which operation to perform. Each of the outputs from the eight 1-bit logic units are then sent to the master mux (figure 2).

The Conditional Unit

The conditional unit is simply responsible for SLT.

The Master Mux

The master MUX is in charge of selecting the final output between the three buses from the other main units. This is done with the help of the 3 select lines.

Control Unit Configuration

The control unit is also very similar to the one designed for lab 5 (figure 6). It is comprised of the following components: Microprogram address decoder, 8-bit ripple carry adder, 3-1 MUX, microcode store, and micro PC. The main components will be further discussed in the subsequent sections.

Instruction Register

The instruction register (IR) is loaded with the contents of the first 4 bits of an instruction from memory (also known as the Opcode).

Microprogram Address Decoder

This component takes in the contents of the IR, concatenates 4 zeros and outputs this value to the 3-1 MUX. This represents the address of the first microinstruction of a given microprogram and is loaded into micro PC when appropriate.

3-1 MUX

The 3-1 MUX dictates the contents of the micro PC for the subsequent clock cycle. Part of the microinstruction layout is dedicated to acting as a select for this MUX. Given this, microinstructions that have a subsequent instruction make the MUX select output from the 8-bit ripple carry adder (PC + 1). The last microinstruction of a microprogram makes the MUX select output "00000000", which allows the control unit to undergo the fetch phase. The very first address of the microcode store is known as the fetch instruction, and makes the MUX output the signal from the Microprogram address decoder (the address of the first micro instruction).

8-bit Ripple Carry Adder

This component simply increments the contents of micro PC by 1, which represents the address of a subsequent micro instruction.

Microcode Store

The microcode store contains the sequence of control bits for every micro program (i.e machine instructions). For the purpose of this project, it will have 3 bits for internal control signals, and 17 control signals for the Datapath.

Changes from Phase 1

Special Register File

Since phase 1, it has become apparent that there is no reason to have 3 bits for the function field in R-type instructions, as control signals for the ALU can be stored directly in the microcode store:

OpCode	Rs	Rt	Rd	<mark>Func</mark>
4 bits	3 bits	3 bits	3 bits	<mark>3 bits</mark>

Given this, it is possible to use these 3 bits for something else. For the updated design, these 3 bits will be used to address registers within a "special" register file (figure 7). These registers will be read-only and will contain the following values: register 0 will contain 0, register 1 will contain 1, register 2 will contain $2^2 = 4$ and so on. The new R-type instruction is the following:

OpCode	Rs	Rt	Rd	Rspecial
4 bits	3 bits	3 bits	3 bits	<mark>3 bits</mark>

In order to access this special register file, Rt will have to be set to "111", thereby replacing the contents going into register B with the specified special register. This is possible through the component "B selector" (figure 1). This means that Rt will not be able to access register 7 in the register file but will be able to access the special register file. Contrary to this, Rs will not be able to access the special register file but will be able to access register 7

Other Special Purpose Registers

Because the overall microprocessor design is based off the MIPS architecture presented in class, there will be the following special purpose registers for storing values between clock cycles: PC IR, MDR, A, B, and ALUout. The contents of PC will point to the current instruction stored in memory and will either be incremented by 2, incremented by 2 + branch, or set to the target field in J type instruction. The instruction register (IR) will store the current instruction that PC is pointing to in memory. Registers A and B will store the values of Rs and Rt/Rspecial respectively. Finally, ALUout will store the output from the ALU. Figure 7 goes into more detail in terms of what these registers are set to on given microinstructions.

General Purpose Registers

In this design, there is a total of eight general purpose registers. Named RO-R7, they reside in the main register file and all have read/write access depending on the register file's control signals (RegWrite).

Appendix

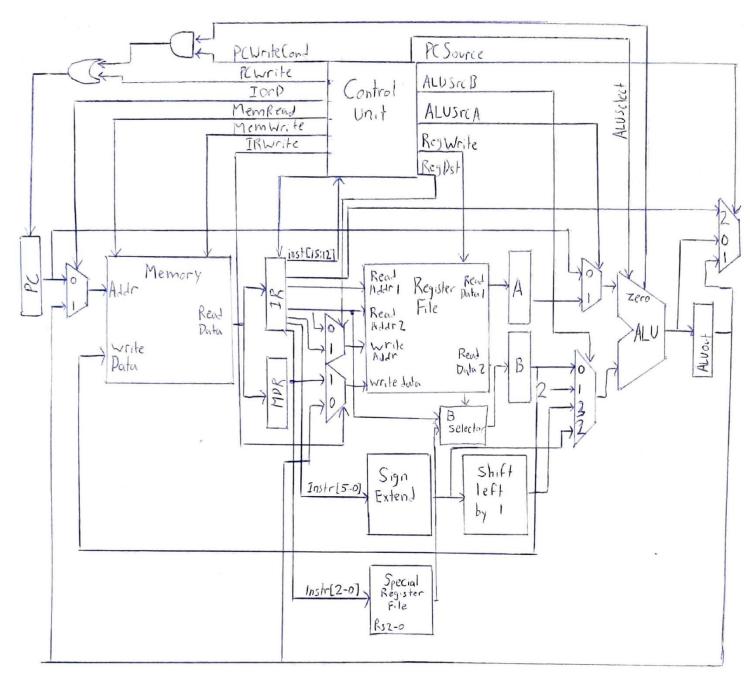


Figure 1: High level block diagram for the microprocessor.

Special Register File

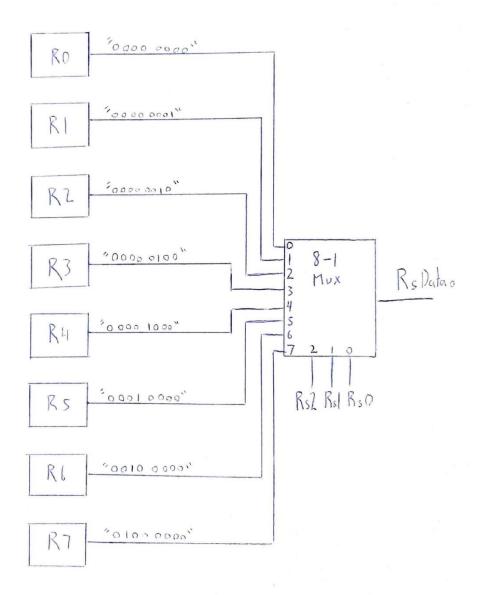


Figure 2: Block diagram for the special register file. These registers are read only. The 8-1 mux outputs the register specified by the Rspecial field.

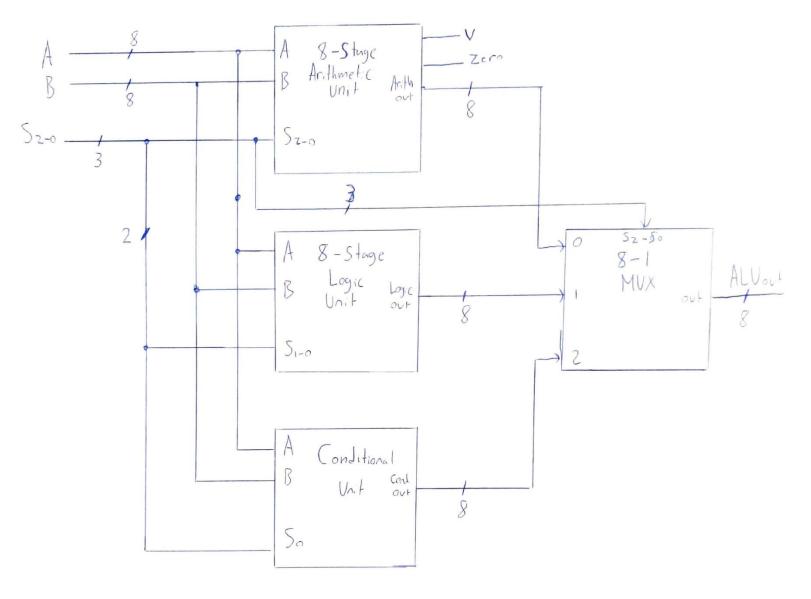


Figure 3: Block diagram for the ALU.

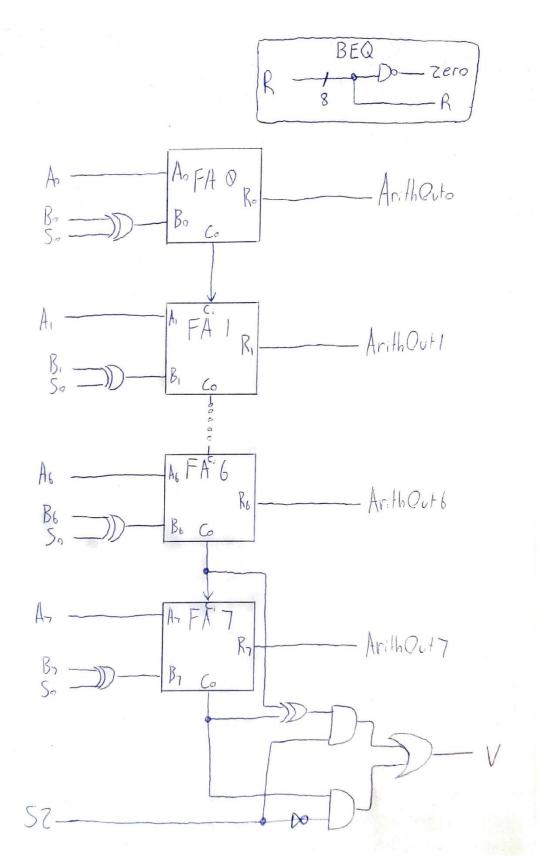


Figure 4: Block diagram for the arithmetic unit. Included is the logic for the zero signal as well as the overflow signal.

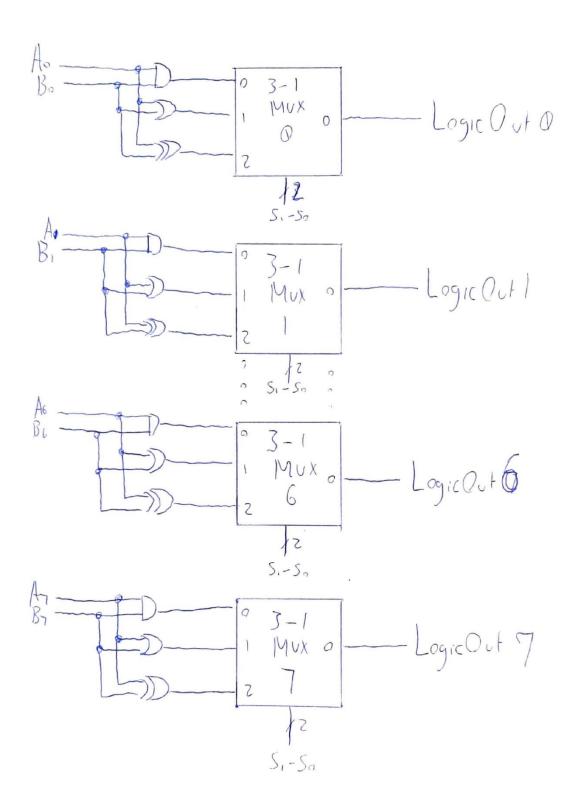


Figure 5: Block diagram for the logic unit.

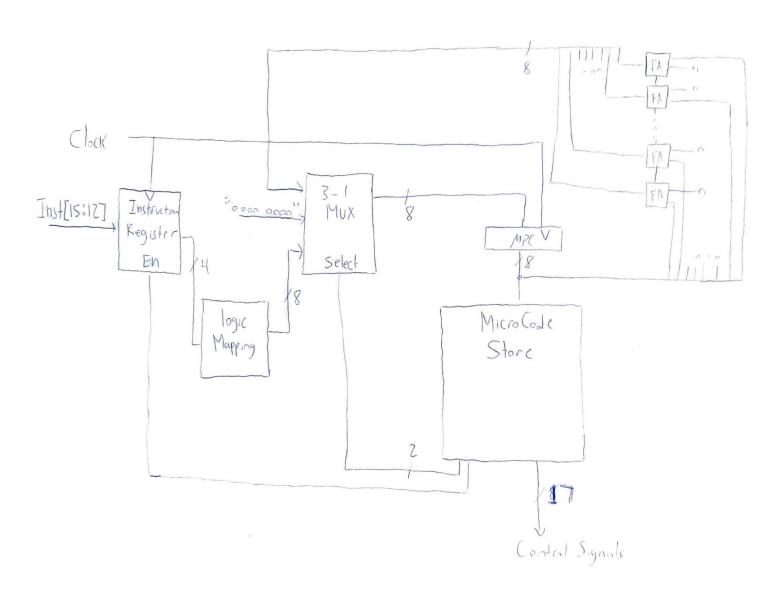


Figure 6: Block diagram for the control unit.

Opcode	Function/Step	Micro IR enable (for		ce ALU OP	ALUSource	eA ALUSourceB	B RegW	∕rite RegDe	est PCWriteCor	d PCW	_	MemRea		rite MemTo	Reg IRWrit
0000	INSTRUCTION FETCH	1	00	010 (add)	0 0	01	o	Х	×	1	o	1	o	Х	1
	DECODE	o	X	010 (add)	0	11	0	х	o	o	X	Х	o	X	o
0001	OR - Compute	0	X	001	1	00	o	X	o	o	X	X	o	x	o
	OR - Store	0	X	XXX	X	XX	1	1	o	o	X	X	o	o	o
0010	Unsigned Add - Compute	0	X	010	1	00	o	X	o	o	Χ	X	o	X	o
	Unsigned Add - Store	0	X	XXX	X	XX	1	1	o	Ó	Χ	X	o	o	o
0011	XOR - Compute	6	X	011	1	00	o	X	ő	o	X	X	o	X	o
	XOR - Store	0	X	XXX	X	XX	1	1	б	o	X	X	o	б	Ő
0100	Signed Add - Compute	o	X	100	1	00	o	Х	ő	o	Х	Х	o	X	o
	Signed Add - Store	o	X	XXX	X	XX	1	1	б	o	X	X	ő	6	6
0101		0	х	101	1	00	o	Х	6	o	х	Х	Ő	х	o
	Signed Subtract - Store	6	×	XXX	х	xx	1	1	6	ő	Х	Х	0	6	6
0110		0	×	110	1	00	o	Х	0	o	Х	Х	o	X	o
		0	х	XXX	Х	XX	1	1	ő	ő	Х	Х	ő	б	6
0111		0	01	111	1	00	6	X	1	o	X	X	0	X	6
1000	AND - Compute	0	X	000	1	00	o	X	6	0	X	X	0	X	0
	AND - Store	6	X	XXX	X	XX	1	î	6	ő	X	X	6	6	6
1001	J.	6	10	XXX	X	XX	6	X	X	1	X	X	6	X	6
1010	ADDIU - Compute	6	X	010	î	10	6	X	ô	ő	X	X	6	X	6
1010	ADDIU - Store	6	X	XXX	X	XX	1	ĥ	ő	6	X	X	6	ô	6
1011	SW - Excecute	6	X	011	^ 1	10	0	X	6	6	X	X	6	X	6
1011		6					6		6	6	'n		1		6
1100	SW - Memory Access	6	X	XXX	X	XX	6	X	6	6		X	6	X	6
1100	ADDI - Compute	o	X	100	1	10	1	X 1	0	0	X	X	0	X	0
	ASSI - Store	6	X	XXX	X	XX	0		0	0	X	X		Ó	
1101	SUBI - Compute		X	101	1	10	1	X			X	X	0	X	0
	SUBI - Store	0	X	XXX	X	XX		1	ō	O	Х	Х	O	Ó	0
1110	SLTI - Compute	0	Х	110	1	10	o	X	o	0	Х	Х	0	X	o
	SLTI - Store	0	X	XXX	X	XX	1	1	6	0	X	X	o	Ó	0
		7				_									
1111	LW - Excecute	o	Х	111	1	10	0	Х	0	0	Х	X	o	Х	o
1111	LW - Memory Access	o o	Х	XXX	1 X	XX	6	Х	ő	6	1	1	ő	Х	б
	LW - Memory Access LW - Write Back	ō ō			1										
pcode	LW - Memory Access LW - Write Back Function/Step	0 Comment	X X	XXX XXX	X X	XX XX	0 1	х б	o	o	1 X	1 X	0	X 1	б б
pcode	LW - Memory Access LW - Write Back Function/Step INSTRUCTION FETCH	0 Comment Read instruction f	X X rom memory (XXX XXX (MemRead =	1 X X = 1), load it	XX XX into IR)IRWrite	0 1 te = 1), inc	X 0 crement PC	o o by 2 (ALUsourc	0 0 e = 01(2)	1 X	1 X	0	X 1	б
Opcode 1000	LW - Memory Access LW - Write Back Function/Step INSTRUCTION FETCH DECODE	0 Comment Read instruction f Read register rs ar	X X rom memory (nd rt into A an	XXX XXX (MemRead = d B, as well a	X X = 1), load it i	XX XX into IR)IRWrite for branch into	0 1 te = 1), inc ALUout I	X 0 crement PC	0 0 C by 2 (ALUsourd s needed for lat	0 0 e = 01(2) er	1 X , ALUOP = 01	1 X	0	X 1	б б
Opcode 1000	LW - Memory Access LW - Write Back Function/Step INSTRUCTION FETCH DECODE OR - Compute	Comment Read instruction f Read register rs ar Perform the speci	X X rom memory (nd rt into A and ified R type ins	XXX XXX (MemRead = d B, as well a struction (us	X X = 1), load it i as address f sing operan	XX XX into IR)IRWrite for branch into ids A and B), sto	te = 1), inco ALUout I	X 0 crement PC IN CASE it i esult into A	0 0 C by 2 (ALUsourd s needed for lat	0 0 e = 01(2) er	1 X , ALUOP = 01	1 X	0	X 1	б б
Opcode 0000	LW - Memory Access LW - Write Back Function/Step INSTRUCTION FETCH DECODE OR - Compute OR - Store	Comment Read instruction f Read register rs ar Perform the speci	X rom memory (nd rt into A and fied R type ins s from ALUout	XXX XXX (MemRead = d B, as well a struction (us	X X = 1), load it is as address f sing operano previous ste	XX XX into IR)IRWrite for branch into ds A and B), sto ep) into the spe	te = 1), inc ALUout I tore the re	X 0 crement PC IN CASE it i esult into A gister	0 0 C by 2 (ALUsourc s needed for lat ALUout regiter fo	0 0 e = 01(2) er or next s	1 X , ALUOP = 01	1 X	0	X 1	o
Opcode 0000 0001	LW - Memory Access LW - Write Back Function/Step INSTRUCTION FETCH DECODE OR - Compute	Comment Read instruction f Read register rs ar Perform the speci	X rom memory (nd rt into A and fied R type ins s from ALUout	XXX XXX (MemRead = d B, as well a struction (us	X X = 1), load it is as address f sing operano previous ste	XX XX into IR)IRWrite for branch into ds A and B), sto ep) into the spe	te = 1), inc ALUout I tore the re	X 0 crement PC IN CASE it i esult into A gister	0 0 C by 2 (ALUsourc s needed for lat ALUout regiter fo	0 0 e = 01(2) er or next s	1 X , ALUOP = 01	1 X	0	X 1	o
Opcode 0000	LW - Memory Access LW - Write Back Function/Step INSTRUCTION FETCH DECODE OR - Compute OR - Store	Comment Read instruction f Read register rs ar Perform the speci	X X from memory (nd rt into A and ffied R type ins s from ALUout ffied R type ins	XXX XXX (MemRead = d B, as well a struction (us truction	X X = 1), load it i as address f sing operan previous ste sing operan	XX XX into IR)IRWrite for branch into ds A and B), sto ep) into the spe ds A and B), sto	te = 1), ind to ALUout I core the re ecified re core the re	X 0 crement PC IN CASE it i esult into A gister esult into A	0 0 C by 2 (ALUsourc s needed for lat ALUout regiter fo	0 0 e = 01(2) er or next s	1 X , ALUOP = 01	1 X	0	X 1	o
Opcode 0000	LW - Memory Access LW - Write Back Function/Step INSTRUCTION FETCH DECODE OR - Compute OR - Store Unsigned Add - Compute	Comment Read instruction f Read register rs ar Perform the speci Store the content Perform the speci	X X rom memory (nd rt into A and fied R type ins s from ALUout fied R type ins s from ALUout	XXX XXX (MemRead = d B, as well a struction (us through the p struction (us through the p struction the p	X X = 1), load it is as address fising operan or evious steep sing operan or evious steep	XX XX into IR)IRWrite for branch into ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe	te = 1), inco ALUout I tore the re ecified re tore the re ecified re	X 6 crement PC IN CASE it i esult into A gister esult into A gister	o by 2 (ALUsource s needed for lat ALUout regiter for ALUout regiter for	o e = 01(2) er or next s	X, ALUOP = 01	1 X	0	X 1	o
Opcode 0000 0001	LW - Memory Access LW - Write Back Function/Step INSTRUCTION FETCH DECODE OR - Compute OR - Store Unsigned Add - Compute Unsigned Add - Store	Comment Read instruction f Read register rs ar Perform the speci Store the content Perform the speci	X X rom memory (nd rt into A and fied R type in: s from ALUout fied R type in: s from ALUout fied R type in:	XXX XXX (MemRead = d B, as well a struction (us c (from the p struction (us c (from the p	X X = 1), load it i as address f sing operan orevious ste sing operan orevious ste sing operan	XX XX into IR) IR Write for branch into ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto	te = 1), inco ALUout I tore the re ecified re ecified re tore the re	X 6 Crement PC IN CASE it i esult into A gister esult into A gister esult into A	o by 2 (ALUsource s needed for lat ALUout regiter for ALUout regiter for	o e = 01(2) er or next s	X, ALUOP = 01	1 X	0	X 1	о О
Dpcode 0000 0001 0001	LW - Memory Access LW - Write Back Function/Step INSTRUCTION FETCH DECODE OR - Compute OR - Store Unsigned Add - Compute Unsigned Add - Store XOR - Compute	Comment Read instruction f Read register rs ar Perform the speci Store the content Perform the speci Store the content Perform the speci	X X rom memory (nd rt into A and fied R type ins s from ALUout fied R type ins s from ALUout fied R type ins	XXX XXX (MemRead = d B, as well a struction (us of the postruction	X X = 1), load it i as address f sing operan orevious ste sing operan orevious ste sing operan orevious ste	XX XX into IR)IRWrite for branch into dds A and B), sto ep) into the spe dds A and B), sto ep) into the spe dds A and B), sto ep) into the spe	te = 1), inco ALUout I core the re ecified re core the re ecified re tore the re ecified re	X 6 Crement PC IN CASE it i esult into A gister esult into A gister esult into A gister	o to y 2 (ALUsources needed for late ALUout regiter for LUOut regiter for L	o o e = 01(2) er or next so or next so	1 X , ALUOP = 01 tep	1 X	0	X 1	б б
Dpcode 0000 0001 0001	LW - Memory Access LW - Write Back Function/Step INSTRUCTION FETCH DECODE OR - Compute OR - Store Unsigned Add - Compute Unsigned Add - Store XOR - Compute XOR - Store Signed Add - Compute	Comment Read instruction f Read register rs ar Perform the speci Store the content Perform the speci Store the content Perform the speci	X X rom memory (nd rt into A and fied R type in: s from ALUout fied R type in: s from ALUout fied R type in: s from ALUout	XXX XXX (MemRead = d B, as well a struction (us c (from the p struction (us c (from the p struction (us	X X =1), load it is as address fising operan orevious steeping operan orevious steeping operan orevious steeping operan orevious steeping operan	XX XX into IR)IRWrite for branch into dds A and B), sto ep) into the spe dds A and B), sto ep) into the spe dds A and B), sto ep) into the spe dds A and B), sto	te = 1), inco o ALUout I core the re ecified re ecified re ecified re ecified re ecified re ecified re	X 6 Crement PC IN CASE it i esult into A gister esult into A gister esult into A gister esult into A gister esult into A	o to y 2 (ALUsources needed for late ALUout regiter for LUOut regiter for L	o o e = 01(2) er or next so or next so	1 X , ALUOP = 01 tep	1 X	0	X 1	o
Dpcode 00000 00011 00111	LW - Memory Access LW - Write Back Function/Step INSTRUCTION FETCH DECODE OR - Compute OR - Store Unsigned Add - Compute Unsigned Add - Store XOR - Compute XOR - Store Signed Add - Compute Signed Add - Compute	Comment Read instruction f Read register is an Perform the speci Store the content	X X rom memory (nd rt into A an- ified R type in: s from ALUout fied R type in: s from ALUout fied R type in: s from ALUout ified R type in: s from ALUout	XXX XXX (MemRead = d B, as well a struction (us to from the p	X X = 1), load it is as address fising operan orevious steeping operan	XX XX into IR)IRWrite for branch into ds A and B), sto ep) into the spends A and B) into the spends A and B).	te = 1), inco ALUout I D ALUout I D ALUot I D ALUout I D ALUot I D ALUout I D ALuot I	X 6 Crement PC IN CASE it i esult into A gister esult into A gister esult into A gister esult into A gister esult into A	o by 2 (ALUsource s needed for lat ALUout regiter for ALUout regiter for ALUout regiter for ALUout regiter for	o o e = 01(2) er or next so or next so or next so	ALUOP = 01 tep tep	1 X	0	X 1	о О
Dpcode 00000 00011 00111	LW - Memory Access LW - Write Back Function/Step INSTRUCTION FETCH DECODE OR - Compute OR - Store Unsigned Add - Compute Unsigned Add - Store XOR - Compute XOR - Store Signed Add - Compute Signed Add - Compute Signed Add - Compute Signed Add - Compute Signed Add - Store Signed Add - Store	Comment Read instruction f Read register rs ar Perform the speci Store the content Perform the speci	X X rom memory (nd rt into A an ified R type in: s from ALUout fied R type in: s from ALUout ified R type in: s from ALUout ified R type in: s from ALUout fied R type in:	XXX XXX (MemRead = d B, as well a struction (us : (from the p struction (us : (from t	1 X X X = 1), load it is as address fising operan or evious steeping operan	XX XX into IR JIRWrite for branch into ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto	te = 1), inco ALUout I ALUout I Core the re	X 6 crement PC IN CASE it i essult into A gister essult into A	o by 2 (ALUsource s needed for lat ALUout regiter for ALUout regiter for ALUout regiter for ALUout regiter for	o o e = 01(2) er or next so or next so or next so	ALUOP = 01 tep tep	1 X	0	X 1	o
Dpcode 0000 0001 00010 00011 0100	LW - Memory Access LW - Write Back Function/Step INSTRUCTION FETCH DECODE OR - Compute OR - Store Unsigned Add - Compute Unsigned Add - Store XOR - Compute XOR - Store Signed Add - Compute Signed Add - Compute Signed Add - Store Signed Add - Store Signed Add - Store Signed Subtract - Compute Signed Subtract - Compute	Comment Read instruction f Read register rs ar Perform the speci Store the content Perform the speci	X X rom memory (nd rt into A ani ified R type in: s from ALUout	XXX XXX (MemRead = d B, as well a struction (us of from the p	1 X X X = 1), load it is as address fising operan or evious steeping o	XX XX into IR JIRWrite for branch into ds A and B), sto ep) into the spe dds A and B), sto ep) into the spe dds A and B), sto ep) into the spe dds A and B), sto ep) into the spe dds A and B), sto ep) into the spe dds A and B), sto ep) into the spe dds A and B), sto ep) into the spe dds A and B), sto ep) into the spe	te = 1), inco o ALUout I done the re- ecified re- toore the re- ecified re- toore the re- ecified re- toore the re- ecified re- toore the re- ecified re-	X 0 crement PC N CASE it i esult into A gister	o by 2 (ALUsource is needed for lat ALUout regiter for all the aluce is a second control of t	o o e = 01(2) er or next so or next so or next so	ALUOP = 01 tep tep tep	1 X	0	X 1	o
Dpcode 0000 0001 00010 00011 0100	LW - Memory Access LW - Write Back Function/Step INSTRUCTION FETCH DECODE OR - Compute OR - Store Unsigned Add - Compute Unsigned Add - Store XOR - Compute XOR - Compute Signed Add - Compute Signed Add - Compute Signed Add - Store Signed Subtract - Compute Signed Subtract - Store SLT - Compute	Comment Read instruction f Read register rs ar Perform the speci Store the content Perform the speci	X X rom memory (nd rt into A ani fied R type ins s from ALUout fied R type ins	XXX XXX (MemRead = d B, as well a struction (us c (from the p struction (us c (from t	X X X = 1), load it is as address fising operan or evious steeping operan	XX XX into IR)IRWrite for branch into ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto	te = 1), inco o ALUout I core the re ecified re core the re ecified re core the re ecified re core the re ecified re core the re	X 0 crement PC N CASE it i esult into A gister esult into A	o by 2 (ALUsource is needed for lat ALUout regiter for all the aluce is a second control of t	o o e = 01(2) er or next so or next so or next so	ALUOP = 01 tep tep tep	1 X	0	X 1	б б
Dpcode 0000 0001 00010 00011 00100	LW - Memory Access LW - Write Back Function/Step INSTRUCTION FETCH DECODE OR - Compute OR - Store Unsigned Add - Compute Unsigned Add - Store XOR - Compute XOR - Store Signed Add - Compute Signed Add - Compute Signed Add - Store Signed Subtract - Compute Signed Subtract - Store SLT - Compute SLT - Store	Comment Read instruction f Read register rs ar Perform the speci Store the content Perform the speci	X X rom memory (nd rt into A ani fied R type ins s from ALUout	XXX XXX (MemRead = d B, as well a struction (us a (from the p struction the p struction (us a (from the p struction the p struction the p struction the p struction (us a (from the p struction the p	1 x x x x x x x x x x x x x x x x x x x	XX XX into IR)IRWrite for branch into ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ids A and B), sto ep) into the spe ids A and B), sto ep) into the spe ids A and B), sto	te = 1), inc o ALUout I core the re- ecified re- tore the re- ecified re- ecified re- tore the re- ecified re-	X 0 crement PC IN CASE it i esult into A gister	To by 2 (ALUsources needed for late NALUout regiter for NALUOUT re	o o e = 01(2) er or next so or next so or next so	ALUOP = 01 tep tep tep	1 X	0	X 1	б б
Dpcode 0000 0001 0010 0011 0100 0111	LW - Memory Access LW - Write Back Function/Step INSTRUCTION FETCH DECODE OR - Compute OR - Store Unsigned Add - Compute Unsigned Add - Store XOR - Compute XOR - Store Signed Add - Compute Signed Add - Store Signed Add - Store Signed Subtract - Compute Signed Subtract - Store SLT - Compute SLT - Store BEQ	Comment Read instruction f Read register rs ar Perform the speci Store the content Perform the speci	X X rom memory (nd rt into A an- ified R type in: s from ALUout contents of ALUout contents of ALUout	XXX XXX (MemRead = d B, as well a struction (us c (from the p us truction (us c (1 x x x x x x x x x x x x x x x x x x x	XX XX into IR JIRWrite for branch into ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe eps ds A and B), sto ep) into the spe eps if the conte	te = 1), inco o ALUout I core the re- ecified re-	x 6 Crement PC IN CASE it i esult into A gister and B are t	To by 2 (ALUsources needed for late ALUout regiter for ALUOUT regiter	o o o o o o o o o o o o o o o o o o o	ALUOP = 01 cep cep cep	1 X 0 (ADDU), P	CSource =	X 1	o
Dpcode 0000 0001 0010 0011 0100 0111	LW - Memory Access LW - Write Back Function/Step INSTRUCTION FETCH DECODE OR - Compute OR - Store Unsigned Add - Compute Unsigned Add - Store XOR - Compute XOR - Compute Signed Add - Compute Signed Add - Store Signed Subtract - Compute Signed Subtract - Store SLT - Compute SLT - Compute SLT - Store BEQ AND - Compute	Comment Read instruction f Read register rs ar Perform the speci Store the content Perform the speci	X X rom memory (nd rt into A an- ified R type in: s from ALUout	XXX XXX (MemRead = d B, as well a struction (us control of the postruction (us control of th	1 x x x x x x x x x x x x x x x x x x x	XX XX into IR) IRWrite for branch into ds A and B), sto ep) into the spe ds A and B) into the spe ds A	te = 1), inco ALUout I tore the re ecified re re and re ecified re ecified re ecified re ents of A alue in the	x 6 Crement PC IN CASE it i esult into A gister esult into A	To by 2 (ALUsources needed for late ALUout regiter for ALUOUT regiter	o o o o o o o o o o o o o o o o o o o	ALUOP = 01 cep cep cep	1 X 0 (ADDU), P	CSource =	X 1	б
Dpcode 10000 10011 10100 10101 10110	LW - Memory Access LW - Write Back Function/Step INSTRUCTION FETCH DECODE OR - Compute OR - Store Unsigned Add - Compute Unsigned Add - Store XOR - Compute XOR - Store Signed Add - Compute Signed Add - Store Signed Add - Store Signed Subtract - Compute Signed Subtract - Store SLT - Compute SLT - Store BEQ	Comment Read instruction f Read register rs ar Perform the speci Store the content Branch using the c Perform the speci	X X rom memory (nd rt into A an ified R type in: s from ALUout ified R type in:	XXX XXX (MemRead = d B, as well a struction (us : (from the p struction (us : (from t	1 x x x x x x x x x x x x x x x x x x x	XX XX into IR) IRWrite for branch into ds A and B), sto ep) into the spe ds A and B) into the spe ds A and B), sto ep) into the spe ds A and B) into the	te = 1), inco ALUout I tore the re ecified re re and re ecified re ecified re ecified re ents of A alue in the	x 6 Crement PC IN CASE it i esult into A gister esult into A	To by 2 (ALUsources needed for late ALUout regiter for ALUOUT Regiter	o o o o o o o o o o o o o o o o o o o	ALUOP = 01 cep cep cep	1 X 0 (ADDU), P	CSource =	X 1	б
Dpcode 10001 10010 10010 10100 10101 10100 10110	LW - Memory Access LW - Write Back Function/Step INSTRUCTION FETCH DECODE OR - Compute OR - Store Unsigned Add - Compute Unsigned Add - Store XOR - Compute XOR - Compute Signed Add - Compute Signed Add - Store Signed Subtract - Compute Signed Subtract - Store SLT - Compute SLT - Compute SLT - Store BEQ AND - Compute	Comment Read instruction f Read register rs ar Perform the speci Store the content Perform the speci	X X rom memory (nd rt into A an ified R type in: s from ALUout ified R type in:	XXX XXX (MemRead = d B, as well a struction (us : (from the p struction (us : (from t	1 x x x x x x x x x x x x x x x x x x x	XX XX into IR) IRWrite for branch into ds A and B), sto ep) into the spe ds A and B) into the spe ds A and B), sto ep) into the spe ds A and B) into the	te = 1), inco ALUout I tore the re ecified re re and re ecified re ecified re ecified re ents of A alue in the	x 6 Crement PC IN CASE it i esult into A gister esult into A	To by 2 (ALUsources needed for late ALUout regiter for ALUOUT Regiter	o o o o o o o o o o o o o o o o o o o	ALUOP = 01 cep cep cep	1 X 0 (ADDU), P	CSource =	X 1	б
Dpcode 00000 00011 0010 0011	LW - Memory Access LW - Write Back Function/Step INSTRUCTION FETCH DECODE OR - Compute OR - Store Unsigned Add - Compute Unsigned Add - Store XOR - Compute XOR - Store Signed Add - Compute Signed Add - Compute Signed Add - Store Signed Add - Store Signed Subtract - Compute Signed Subtract - Store SLT - Compute SLT - Compute BEQ AND - Compute AND - Store	Comment Read instruction f Read register rs ar Perform the speci Store the content Branch using the c Perform the speci	X X rom memory (nd rt into A an ified R type in: s from ALUout contents of ALI fied R type in: s from ALUout contents of IN:	XXX XXX (MemRead = d B, as well a struction (us of from the postruction (us of from t	x x x x x x x x x x x x x x x x x x x	XX XX into IR JIRWrite for branch into IR JIRWrite for branch into Ids A and B), sto p) into the spe Ids A a	te = 1), inco o ALUout I dore the re ecified re core the re ecified re ecified re ecified re ecified re ecified re ecified re ents of A a	x 6 crement PC N CASE it i esult into A gister and B are t e specified gister	by 2 (ALUsource s needed for lat ALUout regiter for	e = 01(2) er or next s	ALUOP = 01 tep tep tep tep tep tep	X 0 (ADDU), P	CSource =	X 1	б б
Dpcode 0000 0001 0001 0011 01100 01101 01110 01111 0000	LW - Memory Access LW - Write Back Function/Step INSTRUCTION FETCH DECODE OR - Compute OR - Store Unsigned Add - Compute Unsigned Add - Store XOR - Compute XOR - Store Signed Add - Compute Signed Add - Compute Signed Add - Store Signed Subtract - Compute Signed Subtract - Compute Signed Subtract - Store SLT - Compute SLT - Store BEQ AND - Compute AND - Store	Comment Read instruction f Read register rs ar Perform the speci Store the content Dear of the content Perform the speci Store the content Content Perform the speci Store the content Load PC with the content	x x x x x x x x x x x x x x x x x x x	XXX XXX (MemRead = d B, as well a struction (us of from the p struction (us of from t	x x x x x x x x x x x x x x x x x x x	XX XX into IR JIRWrite for branch into ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A signextence ds A signextence ds A signextence	te = 1), inc b ALUout I core the re ecified re re	x 6 crement PC N CASE it i esult into A gister esult into A giste	by 2 (ALUsource s needed for lat ALUout regiter for	e = 01(2) er or next s	ALUOP = 01 tep tep tep tep tep tep	X 0 (ADDU), P	CSource =	X 1	б б
00000000000000000000000000000000000000	LW - Memory Access LW - Write Back Function/Step INSTRUCTION FETCH DECODE OR - Compute OR - Store Unsigned Add - Compute Unsigned Add - Store XOR - Compute XOR - Store Signed Add - Store Signed Subtract - Compute Signed Subtract - Store SIT - Compute SLT - Store BEQ AND - Compute AND - Store J ADDIU - Compute	Comment Read instruction f Read register rs ar Perform the speci Store the content Branch using the c Perform the speci Store the content Using the content Co	x x x rom memory (nd rt into A ani ified R type in: s from ALUout contents of AL ified R type in: s from ALUout contents of IN: ified I type ins s from ALUout	XXX XXX (MemRead = d B, as well a struction (us of from the p	x x x x x x x x x x x x x x x x x x x	into IR) IRWrite for branch into ds A and B), sto ep) into the spe dds A and B), sto ep) into the spe dds A and B), sto ep) into the spe dds A and B), sto ep) into the spe dds A and B), sto ep) into the spe dds A and B), sto ep) into the spe dds A and B), sto ep) into the spe dds A and B), sto ep) into the spe dds A and B), sto ep) into the spe dd A and the val ep) into the spe dd A and the val ep) into the spe dd A signextence ep) into the spe ep) into the spe exp) into the spe	te = 1), inc b ALUout I b ALUout I core the re ecified re ecified re core the re ecified re core the re ecified re core the re ecified re ecified re ecified re	X O crement PC N CASE it i esult into A gister ediate), str gister ediate), str gister	by 2 (ALUsource s needed for lat ALUout regiter for	e = 01(2) er or next s	ALUOP = 01 tep tep tep tep tep tep	X 0 (ADDU), P	CSource =	X 1	б
Opcode 0000 0001 0001 0001 0001 0000 0001 0000 0001 0000 0001	LW - Memory Access LW - Write Back Function/Step INSTRUCTION FETCH DECODE OR - Compute OR - Store Unsigned Add - Compute Unsigned Add - Store XOR - Compute XOR - Store Signed Add - Store Signed Add - Store Signed Add - Store Signed Add - Store Signed Subtract - Compute Signed Subtract - Store SLT - Compute SLT - Store BEQ AND - Compute AND - Store J ADDIU - Compute ADDIU - Compute	Comment Read instruction f Read register rs ar Perform the speci Store the content Branch using the Perform the speci Store the content Load PC with the c Perform the speci	x x x x x x x x x x x x x x x x x x x	XXX XXX (MemRead = d B, as well a struction (us a (from the p stee address (1 X X X X X X X X X X X X X X X X X X X	into IR) IRWrite for branch into ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe dd A and the val ep) into the spe ds A signextence ep) into the spe endiate, store in	te = 1), inc to ALUout I to ALUout I to a ALUot I to a ALUout I to a Alu	X O crement PC N CASE it i esult into A gister esult into A gist	by 2 (ALUsource s needed for lat ALUout regiter for	e = 01(2) er or next s	ALUOP = 01 tep tep tep tep tep tep	X 0 (ADDU), P	CSource =	X 1	б
Dpcode 1000 10010	LW - Memory Access LW - Write Back Function/Step INSTRUCTION FETCH DECODE OR - Compute OR - Store Unsigned Add - Compute Unsigned Add - Store XOR - Compute XOR - Store Signed Add - Compute Signed Add - Store Signed Subtract - Compute Signed Subtract - Compute SLT - Compute SLT - Compute SLT - Store BEQ AND - Compute AND - Store J ADDIU - Compute ADDIU - Compute SW - Excecute SW - Memory Access	Comment Read instruction f Read register rs ar Perform the speci Store the content Branch using the c Perform the speci Store the content Compute the addi Store the content	X X rom memory (nd rt into A an- ified R type ins s from ALUout contents of ALI ified I type ins s from ALUout ified I type ins s from ALUout contents of INI ified I type ins s from ALUout ified I type ins	XXX XXX (MemRead = d B, as well a struction (us c (from the p se address (crom previous)	1 X X X X X X X X X X X X X X X X X X X	XX XX into IR) IRWrite for branch into ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and the spe ds A and the val ep) into the spe ds A signextence pp) into the spe	te = 1), inco ALUout I tore the re ecified re core the re ecified re ecified re ecified re ecified re ents of A alue in the ecified re ided imm ecified re it in ALUo tored in r	x 6 Crement PC IN CASE it i esult into A gister ediate esult into A gister ediate esult into A gister esu	by 2 (ALUsources needed for lat ALUout regiter for a ALUout reg	e = 01(2) er or next so	ALUOP = 01 Rep Rep Rep Rep Rep Rep Rep Re	1 X 0 (ADDU), P	CSource =	X 1	б
Dpcode 0000 0001 0001 0001 0010 0100 0110 0110 0111 00111 00100 01	LW - Memory Access LW - Write Back Function/Step INSTRUCTION FETCH DECODE OR - Compute OR - Store Unsigned Add - Compute Unsigned Add - Store XOR - Compute XOR - Store Signed Add - Compute Signed Add - Compute Signed Add - Store Signed Add - Store Signed Subtract - Compute Signed Subtract - Store SLT - Compute SLT - Store BEQ AND - Compute AND - Store J ADDIU - Compute ADDIU - Store SW - Excecute SW - Memory Access ADDI - Compute	Comment Read instruction f Read register rs ar Perform the speci Store the content Compute the speci Store the content Compute the addi Store the content Perform the speci	X X rom memory (nd rt into A an- ified R type ins s from ALUout fied R type ins s from ALUout ified R type ins s from ALUout contents of ALI ified I type ins s from ALUout contents of INI ified I type ins s from ALUout tition of the ba s of ALUout (fr ified I type ins	XXX XXX (MemRead = d B, as well a struction (us a from the p see address (a from previou truction (using tru	1 X X X X X X X X X X X X X X X X X X X	XX XX into IR JIRWrite for branch into IR JIRWrite for branch into Ids A and BJ, stopp) into the speeds A signextence and A and the value point of the speeds A signextence and A and the speeds A signextence and A signey	te = 1), inc b ALUout I core the re ecified re re in the ecifie	crement PC IN CASE it i esult into A gister and B are t e specified gister ediate), ste gister ut egister ediate), ste ediate), ste ediate), ste	by 2 (ALUsources needed for lat ALUout regiter for a ALUout reg	e = 01(2) er or next so	ALUOP = 01 Rep Rep Rep Rep Rep Rep Rep Re	1 X 0 (ADDU), P	CSource =	X 1	б
Dpcode: 0000 0011 0100 0111 1100 0011 0010 0011 1110 0001 0011 1100	LW - Memory Access LW - Write Back Function/Step INSTRUCTION FETCH DECODE OR - Compute OR - Store Unsigned Add - Compute Unsigned Add - Store XOR - Compute XOR - Store Signed Add - Compute Signed Add - Compute Signed Add - Store Signed Add - Store Signed Subtract - Compute Signed Subtract - Store SLT - Compute SLT - Compute AND - Compute AND - Compute AND - Store J ADDIU - Compute ADDIU - Store SW - Excecute SW - Memory Access ADDI - Compute ASSI - Store	Comment Read instruction f Read register rs ar Perform the speci Store the content Compute the content Compute the add Store the content Compute the add Store the content Perform the speci Store the content Compute the add Store the content Perform the speci	x x x rom memory (XXX XXX (MemRead = d B, as well a struction (us of from the postruction (us of from t	1 X X X X X X X X X X X X X X X X X X X	XX XX Into IR JIRWrite for branch into the speeds A and BJ, stop Jinto the speeds A signextend and the value of JIRWrite for branch into the speeds A signextend by Jinto the speeds A signey A si	te = 1), inc b ALUout I core the re ecified re re core the re ecified re ecified re ents of A A alue in the ecified re it in ALUo tored in r ided imm ecified imm ecified re	x 6 crement PC N CASE it i esult into A gister ediate gister ediate ediate), ste gister but egister B.	by 2 (ALUsource is needed for lat ALUout regiter for the same. Special register for the result in the resu	or next soor nex	ALUOP = 01 Rep Rep Rep Rep Rep Rep Rep Re	o ALUout re	CSource =	X 1	o
Dpcode: 0000 0001 0010 0011 0100 0101 0101 01	LW - Memory Access LW - Write Back Function/Step INSTRUCTION FETCH DECODE OR - Compute OR - Store Unsigned Add - Compute Unsigned Add - Store XOR - Compute XOR - Store Signed Add - Compute Signed Add - Compute Signed Add - Store Signed Subtract - Compute Signed Subtract - Compute Signed Subtract - Store SIT - Compute SLT - Store BEQ AND - Compute AND - Store J ADDIU - Compute ADDIU - Compute ADDIU - Store SW - Excecute SW - Memory Access ADDI - Compute ASSI - Store SUBI - Compute	Comment Read instruction f Read register rs ar Perform the speci Store the content Compute the content Compute the doll Store the content Compute the add Store the content Perform the speci Store the content Compute the add Store the content Perform the speci Store the content Perform the speci Store the content	x x x rom memory (nd rt into A an ified R type in: s from ALUout contents of ALI ified R type in: s from ALUout contents of IN: ified I type ins s from ALUout contents of the ba s of ALUout (ified I type ins s from ALUout ified I type ins	XXX XXX (MemRead = d B, as well a struction (us of from the postruction the postruction (us of from the postruction t	x x x x x x x x x x x x x x x x x x x	XX XX into IR JIRWrite for branch into the speeds A and BJ, stopp into the speeds A and BJ, stopp into the speeds A and BJ, stopp into the speeds A signextence in the speeds A signextence in the address stop into the speeds A signextence in the speeds A	te = 1), inc b ALUout I dore the re ecified re core the re ecified re ecified re ecified re ecified re ecified re ecified re core the re ecified re ecified re core the re ecified re ecified re core the re ecified re ecified re inded imm	x 6 crement PC N CASE it i esult into A gister ediate), ste	by 2 (ALUsource is needed for lat ALUout regiter for the same. Special register for the result in the resu	or next soor nex	ALUOP = 01 Rep Rep Rep Rep Rep Rep Rep Re	o ALUout re	CSource =	X 1	б
Dpcode 0000 1001 1000 1010 1010 1010 1011 1011 1000 1011	LW - Memory Access LW - Write Back Function/Step INSTRUCTION FETCH DECODE OR - Compute OR - Store Unsigned Add - Compute Unsigned Add - Store XOR - Compute Signed Add - Store Signed Add - Store Signed Add - Store Signed Add - Store Signed Subtract - Compute Signed Subtract - Store Signed Subtract - Store SIT - Compute SLT - Compute AND - Compute AND - Compute AND - Store J ADDIU - Compute ADDIU - Store SW - Excecute SW - Memory Access ADDI - Compute ASSI - Store SUBI - Compute SUBI - Compute	Comment Read instruction f Read register rs ar Perform the speci Store the content Branch using the c Perform the speci Store the content Compute the add Store the content Perform the speci Store the content Compute the add Store the content Perform the speci Store the content Perform the speci Store the content	x x x rom memory (nd rt into A an- ified R type ins s from ALUout contents of ALI ified I type ins s from ALUout ified I type ins	XXX XXX (MemRead = d B, as well a struction (us of from the postruction (us of from t	x x x x x x x x x x x x x x x x x x x	XX XX Into IR JIRWrite for branch into the speeds A and BJ, stopp into the speeds A and BJ, stopp into the speeds A and BJ, stopp into the speeds A signextence in the speeds A signey in the	te = 1), inc b ALUout I core the re ecified re in the ecified re in the ecified re it in ALUo tored in re ided imm ecified imm ecified imm ecified imm ecified re ided imm ecified re ided imm ecified re ided imm	x 6 crement PC N CASE it i esult into A gister ediate), ste gister	by 2 (ALUsources needed for late ALUout regiter for the same. special register for the same.	e = 01(2) er or next s	ALUOP = 01 Rep Rep Rep Rep Rep Rep Rep Re	o ALUout re	CSource =	X 1	5 5
Dpcode: 0000 0001 0010 0011 0100 0101 0101 01	LW - Memory Access LW - Write Back Function/Step INSTRUCTION FETCH DECODE OR - Compute OR - Store Unsigned Add - Compute Unsigned Add - Store XOR - Compute XOR - Store Signed Add - Store Signed Add - Store Signed Add - Store Signed Add - Store Signed Subtract - Compute Signed Subtract - Store SIT - Compute SUT - Store BEQ AND - Compute AND - Store J ADDIU - Compute ADDIU - Store SW - Excecute SW - Memory Access ADDI - Compute ASSI - Store SUBI - Compute SUBI - Store SUBI - Compute SUBI - Store	Comment Read instruction f Read register rs ar Perform the speci Store the content Branch using the C Perform the speci Store the content Compute the addi Store the content Perform the speci Store the content	x x x x x x x x x x x x x x x x x x x	XXX XXX (MemRead = d B, as well a struction (us of from the postruction (us of from t	x x x x x x x x x x x x x x x x x x x	xx xx xx into IR JIRWrite for branch into the spe IR JIRWrite for bran	te = 1), inc b ALUout I core the re ecified re in the ecified re it in ALUo tored in r ided imm ecified imm ecified me ecified re ided imm	x 6 crement PC N CASE it i esult into A gister ediate), str gister ediate), str gister ediate), str gister ediate), str gister esult into A	by 2 (ALUsources needed for late ALUout regiter for the same. special register for the same.	e = 01(2) er or next s	ALUOP = 01 Rep Rep Rep Rep Rep Rep Rep Re	o ALUout re	CSource =	X 1	б б
Dpcode 1000	LW - Memory Access LW - Write Back Function/Step INSTRUCTION FETCH DECODE OR - Compute OR - Store Unsigned Add - Compute Unsigned Add - Store XOR - Compute XOR - Store Signed Add - Store Signed Add - Store Signed Subtract - Compute Signed Subtract - Store SLT - Compute SLT - Store BEQ AND - Compute AND - Store J ADDIU - Compute ADDIU - Store SW - Excecute SW - Memory Access ADDI - Compute ASSI - Store SUBI - Compute SUBI - Compute SUBI - Compute SUBI - Store SUBI - Compute SUBI - Store SUBI - Compute	Comment Read instruction f Read register rs ar Perform the speci Store the content Branch using the content Dear the content Compute the add Store the content Perform the speci Store the content Compute the add Store the content Perform the speci Store the content Perform the speci Store the content Store the content Perform the speci	x x x rom memory (nd rt into A an- ified R type in: s from ALUout contents of ALI ified R type in: s from ALUout ified I type in: s from ALUout ified I type in: s from ALUout ified I type ins	XXX XXX (MemRead = d B, as well a struction (us control (us contr	x x x x x x x x x x x x x x x x x x x	xx xx xx into IR JIRWrite for branch into IR JIRWrite for branch into Ids A and B), sto Ep) into the spends A and the value into the spends A signextence in the address sto A signextence in the spends A and B), sto Ep) into the spends A and B).	te = 1), inc o ALUout I core the recified recore the	crement PC IN CASE it i esult into A gister ediate ediate), ste gister esult into A gister ediate), ste gister ediate), ste gister esult into A gister esult into A gister	by 2 (ALUsources needed for late ALUout regiter for the same. special register for the same.	e = 01(2) er or next s	ALUOP = 01 Rep Rep Rep Rep Rep Rep Rep Re	o ALUout re	CSource =	X 1	o
00000000000000000000000000000000000000	LW - Memory Access LW - Write Back Function/Step INSTRUCTION FETCH DECODE OR - Compute OR - Store Unsigned Add - Compute Unsigned Add - Store XOR - Compute XOR - Store Signed Add - Store Signed Add - Store Signed Add - Store Signed Add - Store Signed Subtract - Compute Signed Subtract - Store SIT - Compute SUT - Store BEQ AND - Compute AND - Store J ADDIU - Compute ADDIU - Store SW - Excecute SW - Memory Access ADDI - Compute ASSI - Store SUBI - Compute SUBI - Store SUBI - Compute SUBI - Store	Comment Read instruction f Read register rs ar Perform the speci Store the content Branch using the C Perform the speci Store the content Compute the addi Store the content Perform the speci Store the content	x x x rom memory (nd rt into A an- ified R type in: s from ALUout contents of ALI ified R type in: s from ALUout ified I type in: s from ALUout ified I type in: s from ALUout ified I type ins	XXX XXX (MemRead = d B, as well a struction (us control (us contr	x x x x x x x x x x x x x x x x x x x	xx xx xx into IR JIRWrite for branch into IR JIRWrite for branch into Ids A and B), sto Ep) into the spends A and the value into the spends A signextence in the address sto A signextence in the spends A and B), sto Ep) into the spends A and B).	te = 1), inc o ALUout I core the recified recore the	crement PC IN CASE it i esult into A gister ediate ediate), ste gister esult into A gister ediate), ste gister ediate), ste gister esult into A gister esult into A gister	by 2 (ALUsources needed for late ALUout regiter for the same. special register for the same.	e = 01(2) er or next s	ALUOP = 01 Rep Rep Rep Rep Rep Rep Rep Re	o ALUout re	CSource =	X 1	б б
00000000000000000000000000000000000000	LW - Memory Access LW - Write Back Function/Step INSTRUCTION FETCH DECODE OR - Compute OR - Store Unsigned Add - Compute Unsigned Add - Store XOR - Compute XOR - Store Signed Add - Store Signed Add - Store Signed Subtract - Compute Signed Subtract - Store SLT - Compute SLT - Store BEQ AND - Compute AND - Store J ADDIU - Compute ADDIU - Store SW - Excecute SW - Memory Access ADDI - Compute ASSI - Store SUBI - Compute SUBI - Compute SUBI - Compute SUBI - Store SUBI - Compute SUBI - Store SUBI - Compute	Comment Read instruction f Read register rs ar Perform the speci Store the content Branch using the content Dear the content Compute the add Store the content Perform the speci Store the content Compute the add Store the content Perform the speci Store the content Perform the speci Store the content Store the content Perform the speci	X X rom memory (nd rt into A an- ified R type in: s from ALUout ified I type in:	XXX XXX (MemRead = d B, as well a struction (us a from the p se address (a from the p struction (us a from the p struction the p struction (us a from the p struction the p struction (us a from the p struction	X X X X X X X X X X X X X X X X X X X	XX XX into IR) IRWrite for branch into ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A and B), sto ep) into the spe ds A signextence ep) into the spe endiate, store in the spe	te = 1), inc b ALUout I tore the re ecified re re in the ecified re it in ALUo tored in re ided imm ecified re ided imm ecified re ided imm ecified re	crement PC IN CASE it i esult into A gister ediate ediate), ste gister esult into A gister ediate), ste gister ediate), ste gister esult into A gister esult into A gister	by 2 (ALUsources needed for late ALUout regiter for the same. special register for the same.	e = 01(2) er or next s	ALUOP = 01 Rep Rep Rep Rep Rep Rep Rep Re	o ALUout re	CSource =	X 1	о О

Figure 7: Contents of the microcode store (above). Comments outlining the function of each micro instruction (below).