Breve descripción del planteamiento seguido por la bibliografía de la materia

Facundo Suarez

Departamento de Electrónica
Universidad Nacional de San Luis

Contenido

Universidad Nacional de San Luis

Consigna

La propuesta es construir una versión simple del Datapath del procesador, el cual sea lo suficientemente capaz de ejecutar un subconjunto reducido del set de instrucciones RISC-V.

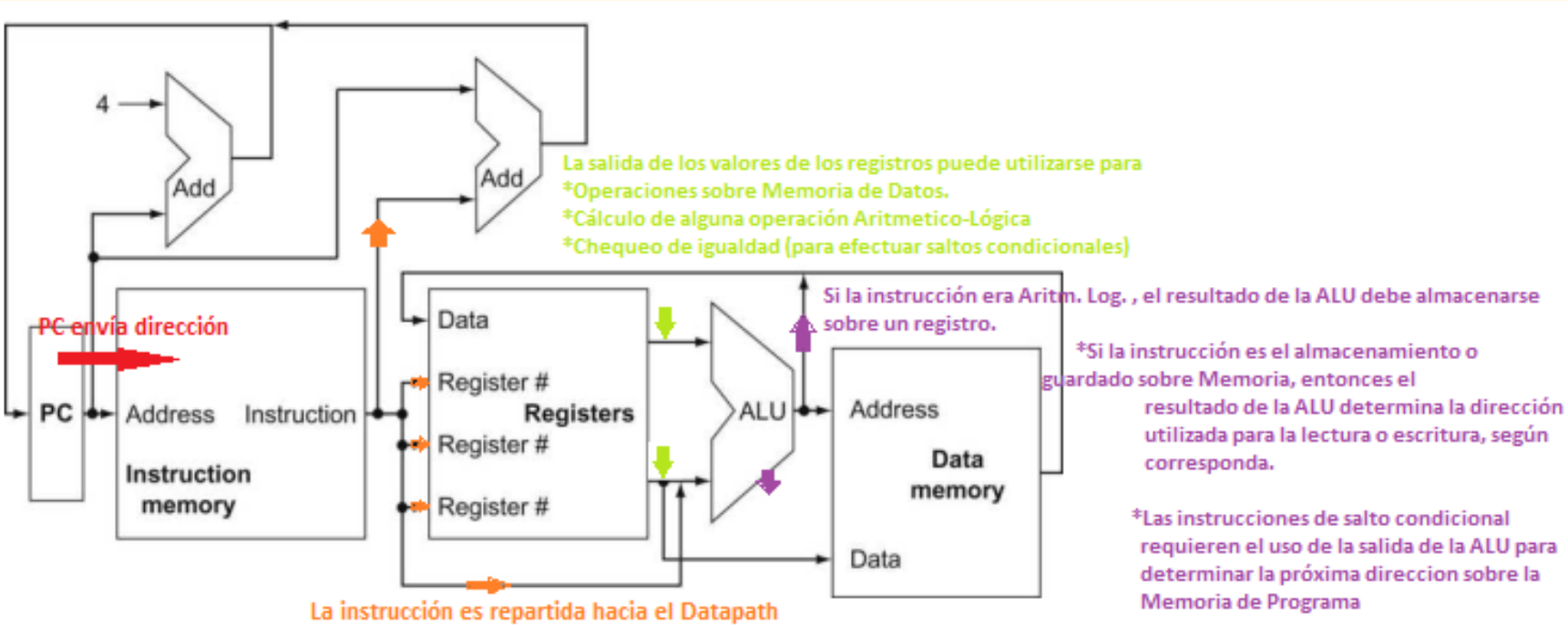


Figura: Vista abstracta de la implementación de un subconjunto de RISC-V, mostrando las principales unidades.

Sobre dicha figura, se omiten dos aspectos principales...

Universidad Nacional de San Luis

- Se omiten los multiplexores que realizan el direccionamiento de los datos que se precisan, dependiendo de la instrucción.
- Se omite la prescencia de una Unidad de Control, que determina las señales de control sobre las unidades principales y de los multiplexores.
  - El multiplexor que determina la dirección de salto condicional es seteado en base a la salida *Zero* de la ALU

Universidad Nacional de San Luis

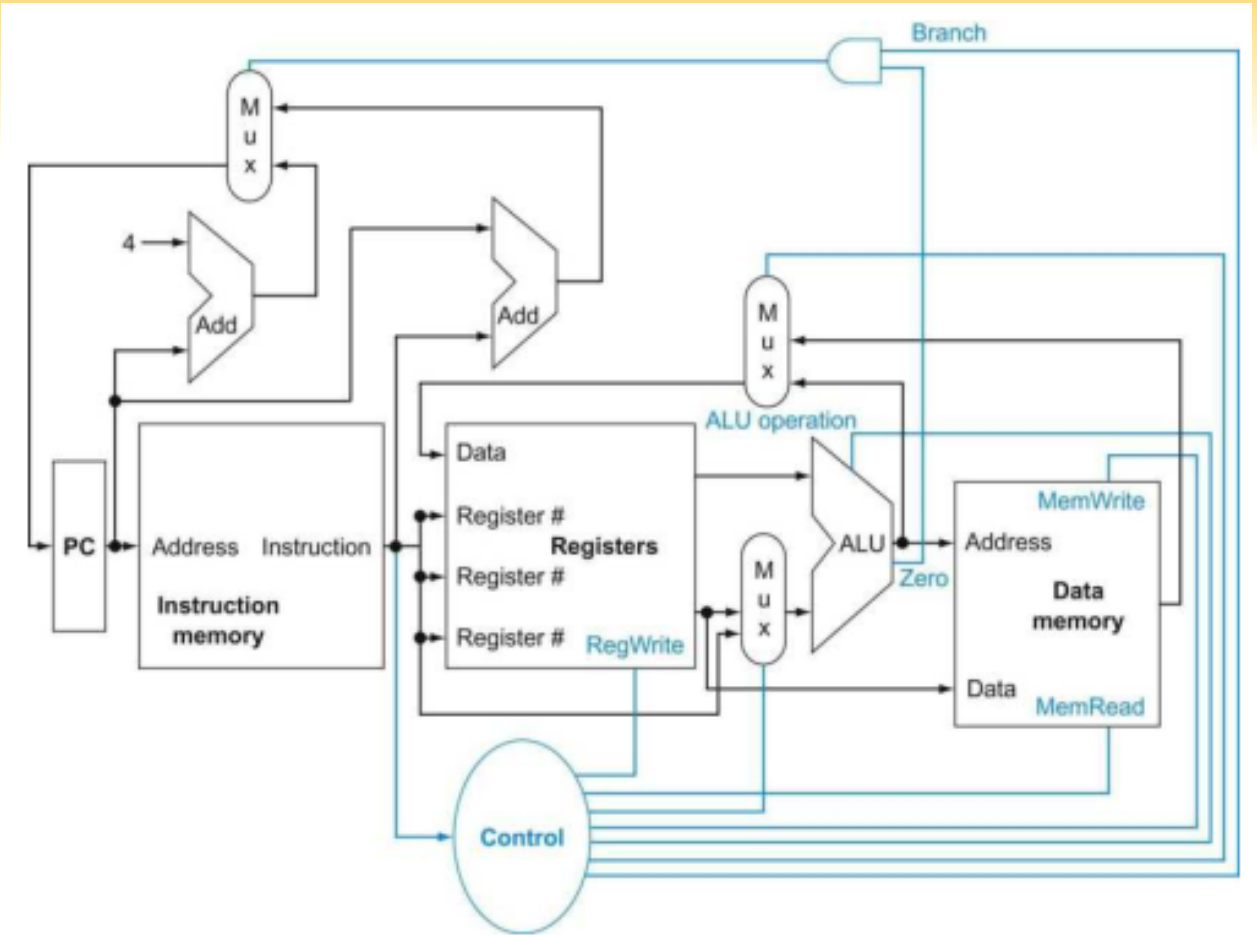


Figura: Vista de la implementación de un subconjunto de RISC-V, mostrando las principales unidades, incluyendo los multiplexores necesarios y las lineas de control.

Universidad Nacional de San Luis

Construcción de Datapath

Una forma de empezar la construcción es examinar los componentes más importantes que requiere el subconjunto del set RISC-V.

- **Program Counter**:Unidad que indica dirección de la instrucción del programa a ejecutar.
  También debe considerarse un sumador que ejecute el incremento de la PC.
- **Memoria de Programa)**: Para la extracción de la instrucción a realizar.

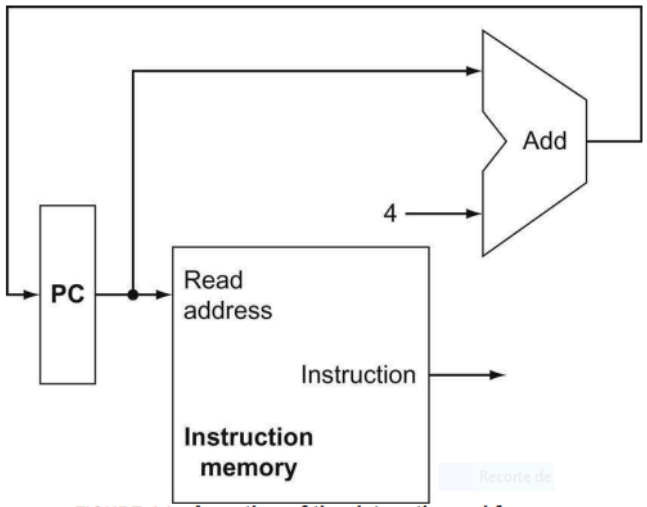


FIGURE 4.6 **A portion of the datapath used for fetching instructions and incrementing the program counter.**
The fetched instruction is used by other parts of the datapath.

Figura: Porción del Datapath.Indicación de PC y la instrucción.

Universidad Nacional de San Luis

- **Registros**: Elemento de estado que consiste en un grupo de registros que pueden ser leídos y escritos a través de una dirección. La mayoría de las operaciones aritmetico-logicas trabaja sobre dichos registros.
- **Memoria de Datos**: Para el manejo de información, las direcciones se calculan a través de la suma con el contenido de un registro.
    - **Extensor de signo**: Unidad encargada de extender un dato representado sobre 12 bits, en un dato del tipo *signed* de 64 bits.

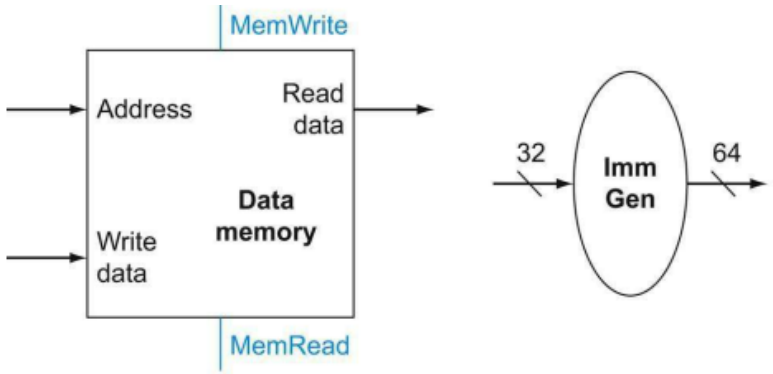


Figura: Porción del Datapath.Carga y guardado sobre memoria. Unidad de extension signo utilizada sobre branches.

Universidad Nacional de San Luis

- Para efectuar los saltos se precisa la unidad de extensión de signo y un sumador. La comparación puede realizarse con la ALU (utilización de la señal *Zero*).

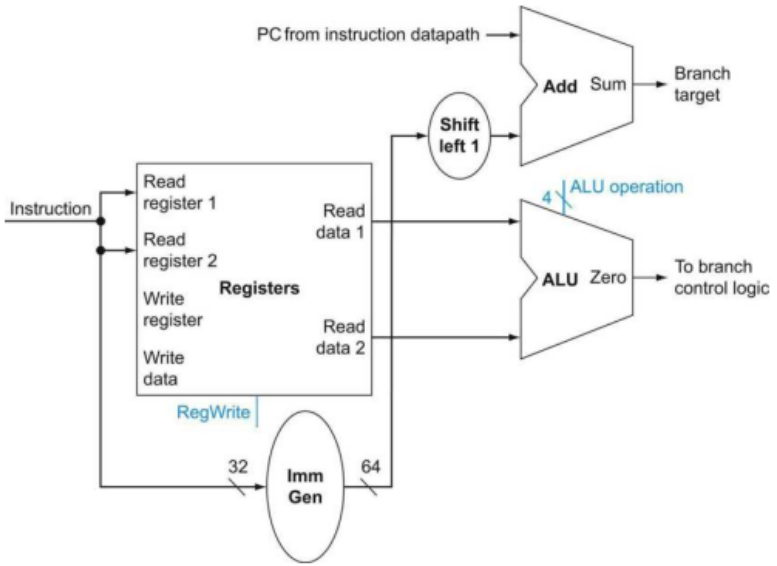


Figura: Porción del Datapath.Sumador y utilización de ALU para efectuar saltos.

Combinando las distintas "secciones"del Datapath, se plantea la inclusión de la Unidad de Control, la cuál esta capacitada para escribir sobre las habilitaciones de escritura, los selectores de cada multiplexor y el control de la ALU.

Universidad Nacional de San Luis

Dado la diferencia entre las señales de control de la ALU y las demas señales, es útil enfocarse en primer lugar sobre ellas. Dependiendo de la clase de la instrucción se realizan una de las siguientes cuatro instrucciones

- **AND**, cuyas lineas de control adquieren valor 0000
- **OR**, cuyas lineas de control adquieren valor 0001
- **ADD**, cuyas lineas de control adquieren valor 0010
- **SUB**, cuyas lineas de control adquieren valor 0110

La unidad de control de 4 bits de la ALU se puede realizar mediante el ingreso de los campos *funct7* y *funct3* de la instrucción.
Su salida sera una señal de 4 bits que adquiere los valores descriptos anteriormente.

Universidad Nacional de San Luis

Instruction opcode	ALUOp	Operation	Funct7 field	Funct3 field	Desired ALU action	ALU control input
ld	00	load doubleword	XXXXXXX	XXX	add	0010
sd	00	store doubleword	XXXXXXX	XXX	add	0010
beq	01	branch if equal	XXXXXXX	XXX	subtract	0110
R-type	10	add	0000000	000	add	0010
R-type	10	sub	0100000	000	subtract	0110
R-type	10	and	0000000	111	AND	0000
R-type	10	or	0000000	110	OR	0001

Figura: Indicación de los valores de control de la ALU, dependiendo tanto de los valores de ALUOp, como de los diferentes opcodes.

Dicha unidad combinacional debería ser capaz de responder ante esta tabla de verdad:

ALUOp		Funct7 field							Funct3 field			
ALUOp1	ALUOp0	I[31]	I[30]	I[29]	I[28]	I[27]	I[26]	I[25]	I[14]	I[13]	I[12]	Operation
0	0	X	X	X	X	X	X	X	X	X	X	0010
X	1	X	X	X	X	X	X	X	X	X	X	0110
1	X	0	0	0	0	0	0	0	0	0	0	0010
1	X	0	1	0	0	0	0	0	0	0	0	0110
1	X	0	0	0	0	0	0	0	1	1	1	0000
1	X	0	0	0	0	0	0	0	1	1	0	0001

Figura: Tabla de verdad de la unidad de control de la ALU.

Diseño de Unidad Control

Revisando los formatos de las cuatro clases de instrucción:

Name (Bit position)	Fields 31:25	24:20	19:15	14:12	11:7	6:0
(a) R-type	funct7	rs2	rs1	funct3	rd	opcode
(b) I-type	immediate[11:0]		rs1	funct3	rd	opcode
(c) S-type	immed[11:5]	rs2	rs1	funct3	immed[4:0]	opcode
(d) SB-type	immed[12,10:5]	rs2	rs1	funct3	immed[4:1,11]	opcode

Figura: Formatos de las instrucciones.

Universidad Nacional de San Luis

Pueden diferenciarse algunos campos interesantes:

- **Opcode**: Ubicado sobre bits de instruccion 6:0.
- **Rs1**: Primer operador de registro para instrucciones tipo R e instrucciones de salto. Tambien especifica la direccion de base para carga y almacenado.
- **Rs2**: Segundo operador de registro, utilizado para instrucciones tipo R e instrucciones de salto.
- Operando de 12 bits utilizado para saltos e instrucciones de guardado.
- **Rd**: Registro de destino para instrucciones tipo R e instrucciones de carga.

Usando esta informacion se reparte los 32 bits de la instrucción sobre el Datapath.

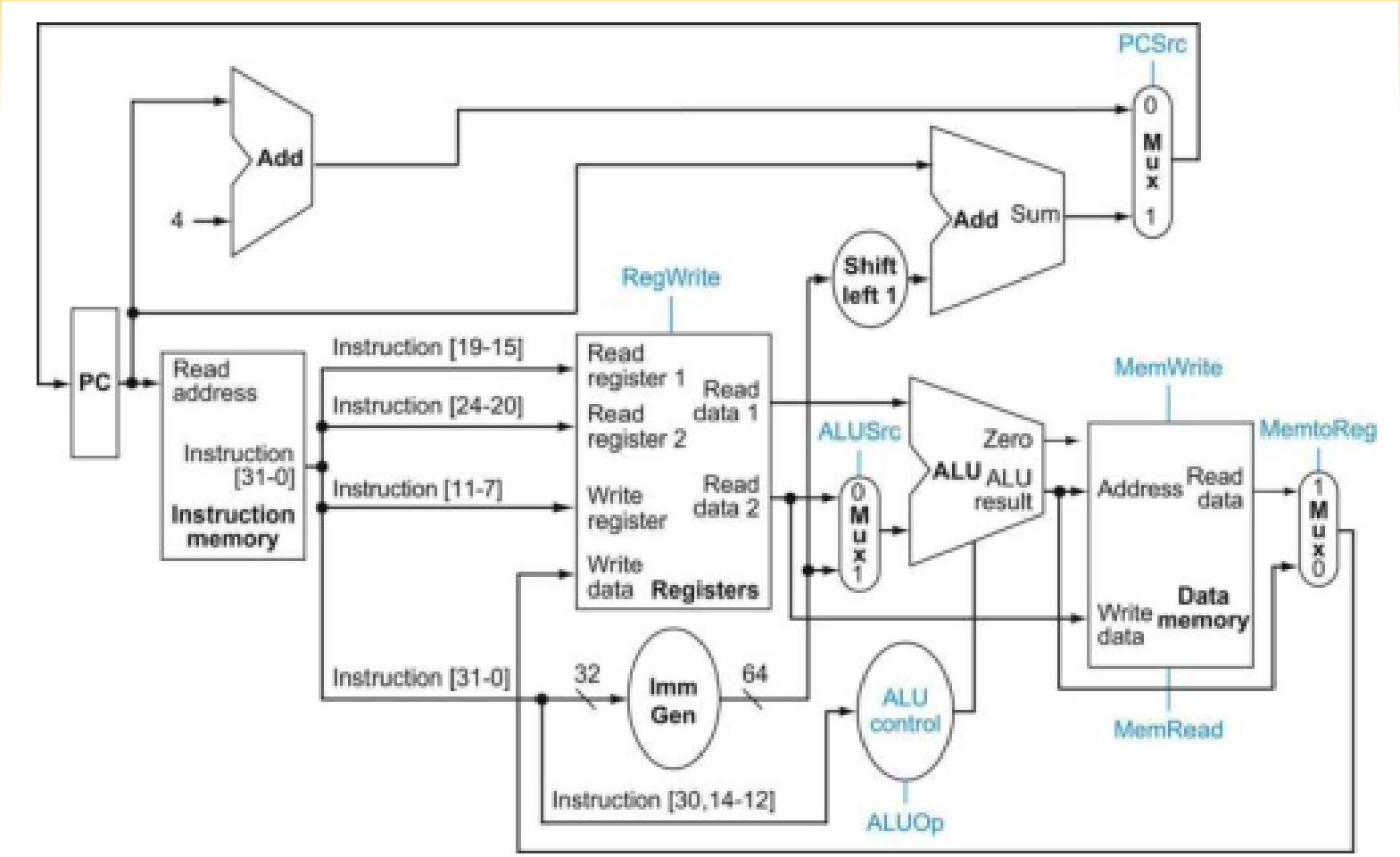


Figura: Versión "parcial"del Datapath.

Universidad Nacional de San Luis

Análisis de señales de control restantes

Una vez definido como trabaja la señal ALUOp, es útil definir qué realizan las demás señales de control.

NOMBRE DE LA SEÑAL

EFECTO CUANDO NO ESTA AFIRMADA LA SEÑAL

EFECTO CUANDO LA SEÑAL ESTA AFIRMADA (o sea es 'verdadero')

Signal name	Effect when deasserted	Effect when asserted
RegWrite	None.	The register on the Write register input is written with the value on the Write data input.
ALUSrc	The second ALU operand comes from the second register file output (Read data 2).	The second ALU operand is the sign-extended, 12 bits of the instruction.
PCSrc	The PC is replaced by the output of the adder that computes the value of PC + 4.	The PC is replaced by the output of the adder that computes the branch target.
MemRead	None.	Data memory contents designated by the address input are put on the Read data output.
MemWrite	None.	Data memory contents designated by the address input are replaced by the value on the Write data input.
MemtoReg	The value fed to the register Write data input comes from the ALU.	The value fed to the register Write data input comes from the data memory.

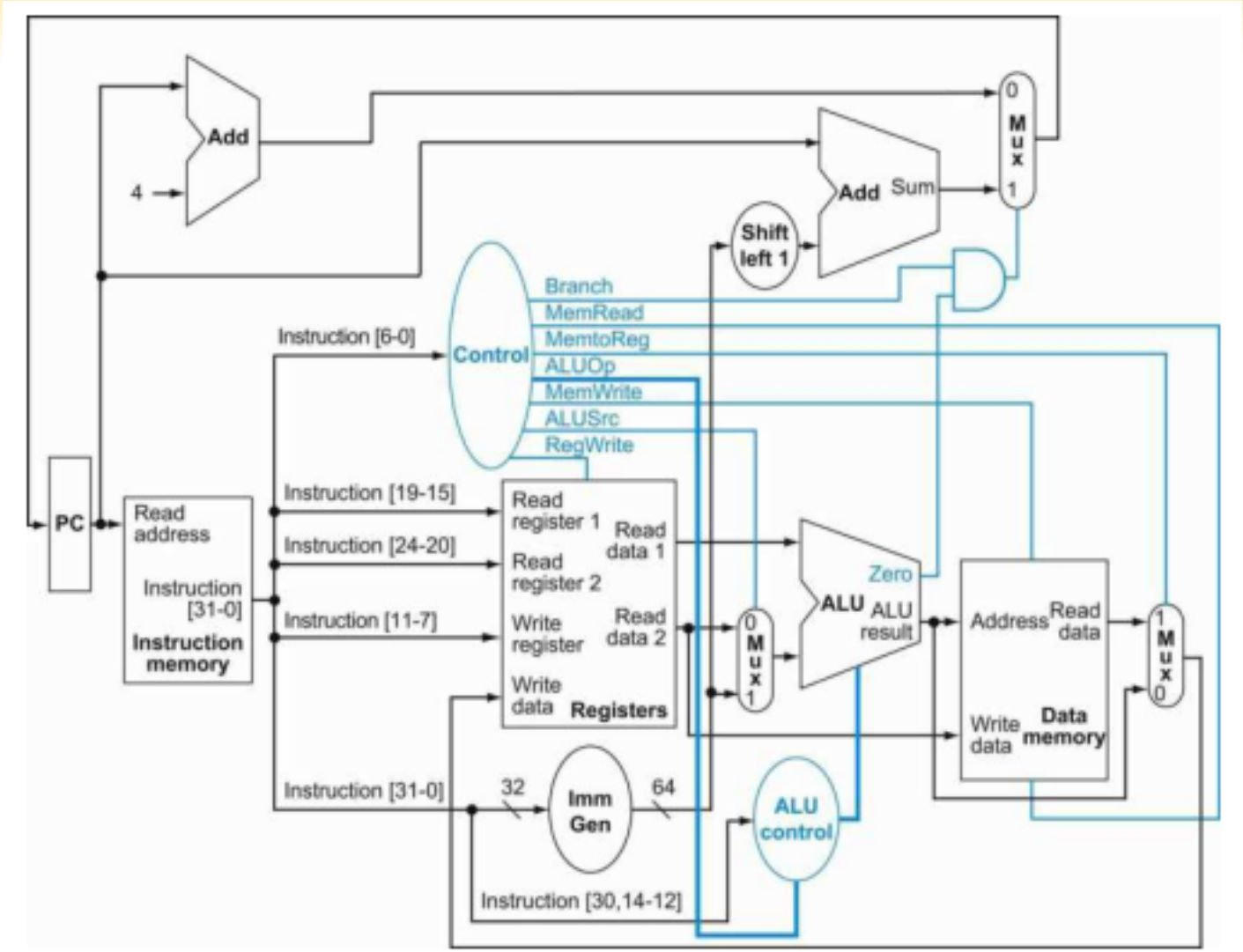
Figura: Función que cumplen las restantes señales de control.

Universidad Nacional de San Luis

Podemos implementar la Unidad de Control, basandose solamente en el opcode y los campos *funct* de la instrucción. Salvo la línea *PCSrc*, dicha linea de control se debe *afirmar*(valor verdadero) si la instrucción usada es un salto condicional. En caso que *Zero*='1' deberá efectuarse el salto. Esto implica la utilización de una AND.

Universidad Nacional de San Luis

La siguiente imagen ilustra el Datapath resultante junto con la Unidad de Control resultante y el Control de la ALU.



Universidad Nacional de San Luis

Figura: Datapath resultante junto su Unidad de Control y la Unidad de Control

Referencias

David A. Patterson, John L. Hennessy.
Computer Organization and Design, RISC-V Edition
Capitulo 4.
Morgan-Kauffman 2017.

Universidad Nacional de San Luis