Diseño Procesador Monociclo

Procesador Monociclo

- Especificación de la arquitectura del repertorio de instrucciones y de las instrucciones que podrá ejecutar el procesador.
- Modelo Carga-Almacenamiento.

Procesador Monociclo

- Se implementará el procesador considerando las instrucciones:
  - Suma, resta y slt (R)
    - add rd, rs, rt
    - sub rd, rs, rt
    - slt rd, rs, rt
  - Or inmediato (I)
    - ori rt, rs, inm16

Procesador Monociclo

- Se implementará el procesador considerando las instrucciones:
  - Carga y Almacenamiento (I).
    - lw rt, inm16(rs)
    - sw rt, inm16(rs)
  - Bifurcación (I).
    - beq rs, rt, rótulo
  - Salto incondicional. (J)
    - j rótulo

Procesador Monociclo

- Manual de Programación MIPS reducido.

Códigos Binarios Assembler.

OP	Rs	Rt	Rd	Sham t	Funct		Nem o	Campo 1	Campo 2	Camp 3	Descripción
000000	Fte1	Fte2	Dst	00000	100000	R	**add**	Dst,	Fte1,	Fte2	#Addition
000000	Fte1	Fte2	Dst	00000	100010	R	**sub**	Dst,	Fte1,	Fte2	#Subtract
000000	Fte1	Fte2	Dst	00000	101010	R	**slt**	Dst,	Fte1,	Fte2	#Set Less Than
000010	jmp26					J	**j**	jmp26			#Jump
000100	Fte1	Fte2	label16			I	**beq**	Fte1,	Fte2,	label16	#Branch on Equal
001101	Fte1	Dst	inm16			I	**ori**	Dst,	Fte1,	inm16	#OR Immediate
100011	Rbase	Dst	offset16			I	**lw**	Dst,	Offset16(RBase)		#Load Word
101011	Rbase	Fte1	offset16			I	**sw**	Fte1,	Offset16(RBase)		#Store Word

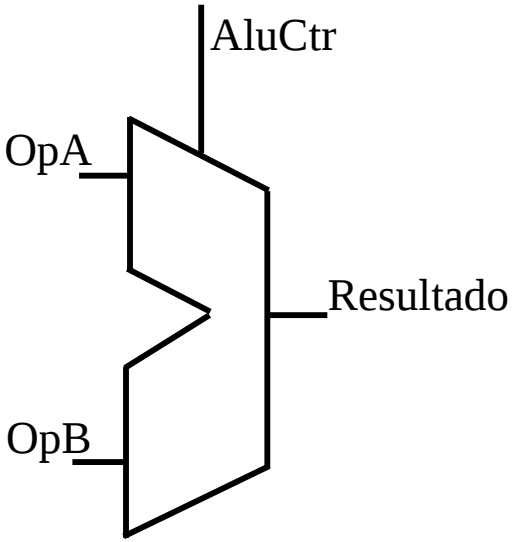
Procesador Monociclo

- Trasferencias Lógicas
  - **add** R[rd] = R[rs] + R[rt]; PC = PC + 4
  - **sub** R[rd] = R[rs] – R[rt]; PC = PC + 4
  - **slt** R[rd] = R[rs] < R[rt] ? 1: 0 ; PC = PC + 4
  - **ori** R[rt] = R[rs] or zero_ext(Inm16); PC = PC + 4
  - **lw** R[rt] = MEM[R[rs] + sign_ext(Inm16)]; PC = PC + 4
  - **sw** MEM[R[rs] + sign_ext(Inm16)] = R[rt]; PC = PC + 4
  - **beq** if (R[rs] = = R[rt])
    PC = (PC +4)+[sign_ext(Inm16)]*4; else PC = PC + 4
  - **j** PC = (PC +4)&0xF0000000 + add_26*4

Procesador Monociclo

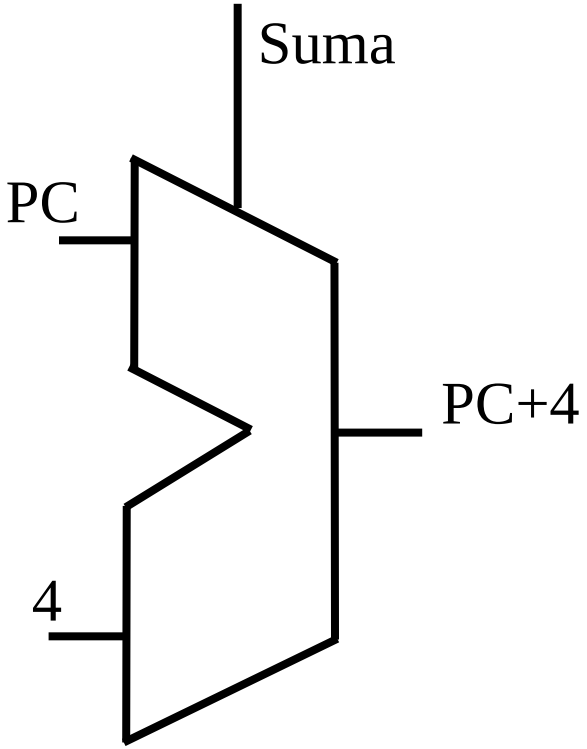
- Recursos Combinacionales
  - Unidad Aritmético Lógica

AluCtr[2..0]		Resultado	Función ALU
Binvert	Operación		
0	00	OpA & OpB	and
0	01	OpA | OpB	or
0	10	OpA + OpB	add
1	10	OpA - OpB	sub
1	11	OpA < OpB ? 1: 0	slt



Procesador Monociclo

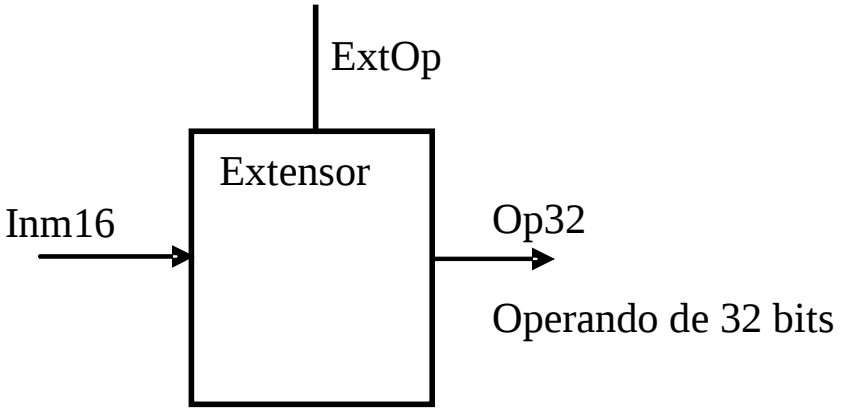
- Recursos Combinacionales
  - Sumador que permita calcular PC+4



Procesador Monociclo

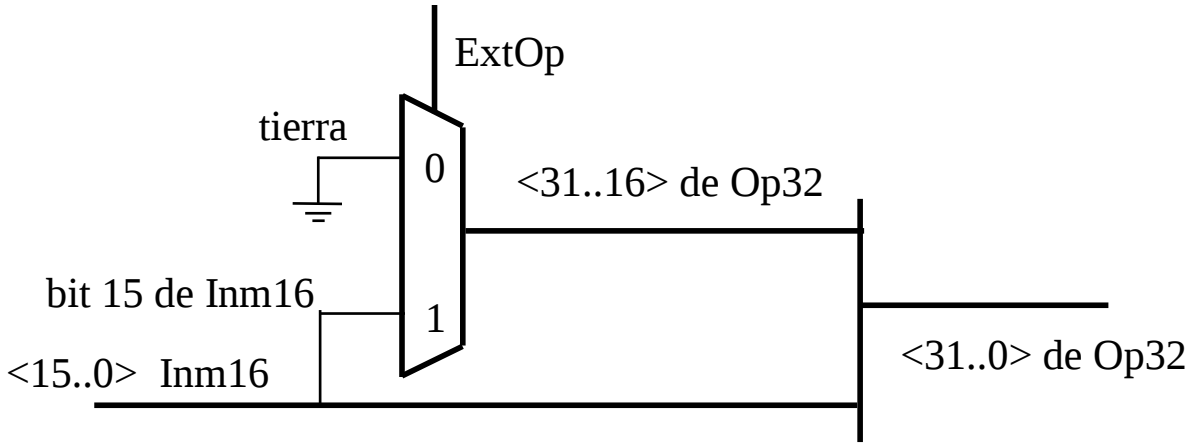
- Recursos Combinacionales
  - Unidad Extensora

ExtOp	Op. de 32
0	zero_ext(Inm16)
1	sign_ext(Inm16)



Procesador Monociclo

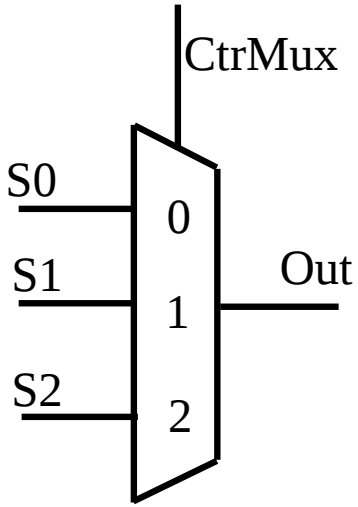
- Recursos Combinacionales
  - Diseño Extensor lógico y aritmético



Procesador Monociclo

- Recursos Combinacionales
  - Multiplexores

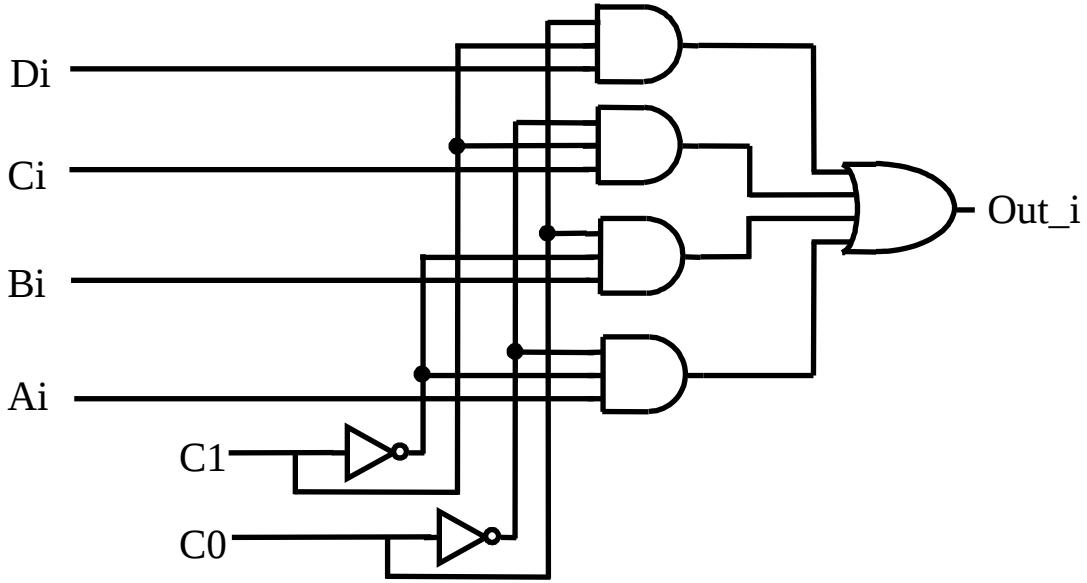
CtrMux	Out
00	S0
01	S1
10	S2



Procesador Monociclo

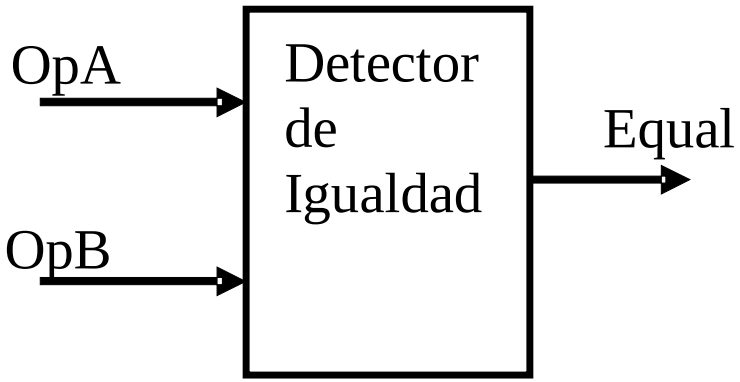
- Recursos Combinacionales
  - Multiplexores. Implementación.

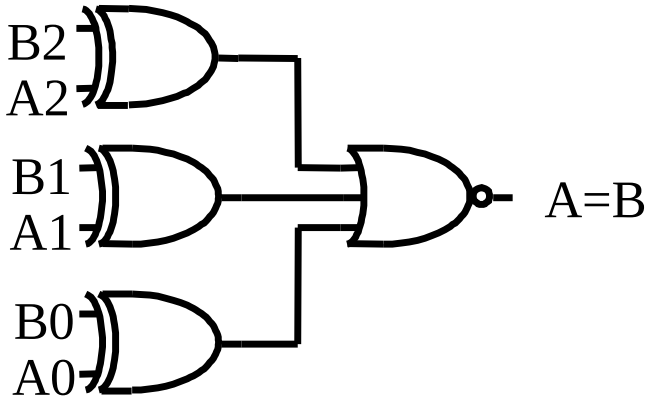
C1	C0	Out_i
0	0	Ai
0	1	Bi
1	0	Ci
1	1	Di



Procesador Monociclo

- Recursos Combinacionales
  - Detector Igualdad



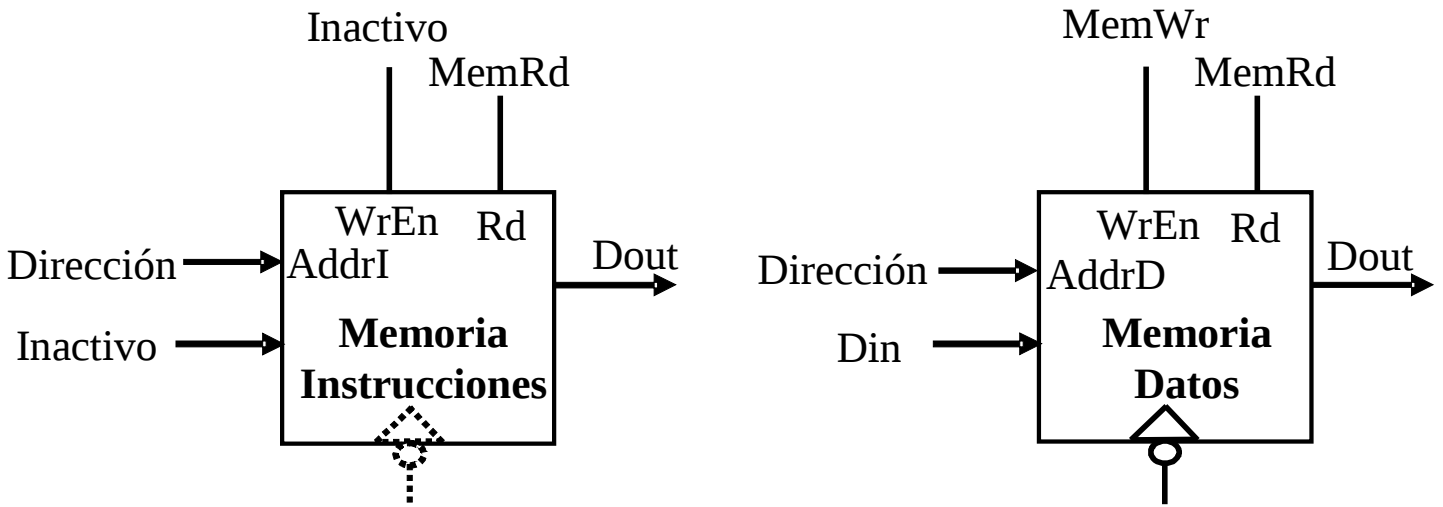


Procesador Monociclo

- Recursos Almacenamiento.
  - Memorias
    - Una memoria para almacenar las instrucciones (ROM).
    - Una memoria para leer y escribir datos (RAM).
    - Diseño inicial se elige tener recursos separados aunque podría ocuparse una sola memoria para datos e instrucciones.

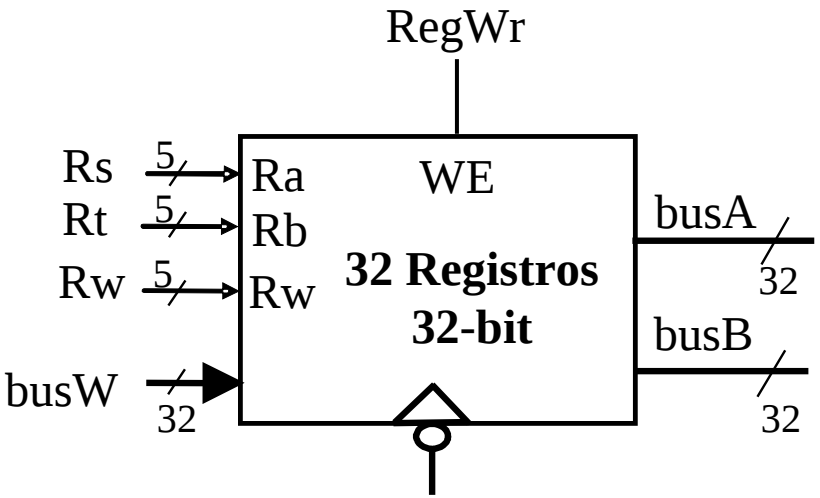
Procesador Monociclo

- Recursos Almacenamiento.
  - Memorias



Procesador Monociclo

- Recursos Almacenamiento.
  - Registros. Arreglo Registros



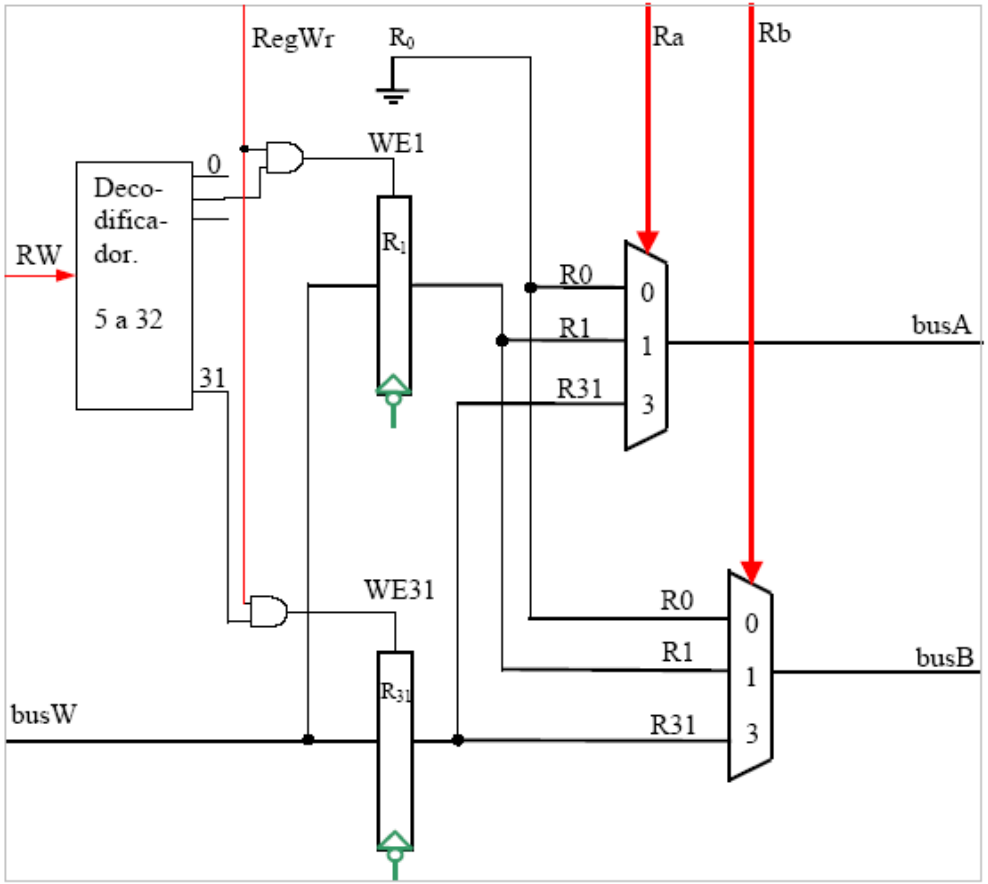
R[Rw] = busW

busA = R[Ra]

busB = R[Rb]

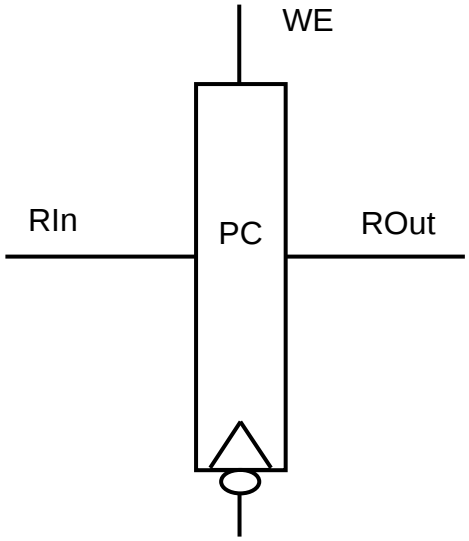
Procesador Monociclo

- Diseño del arreglo de Registros.



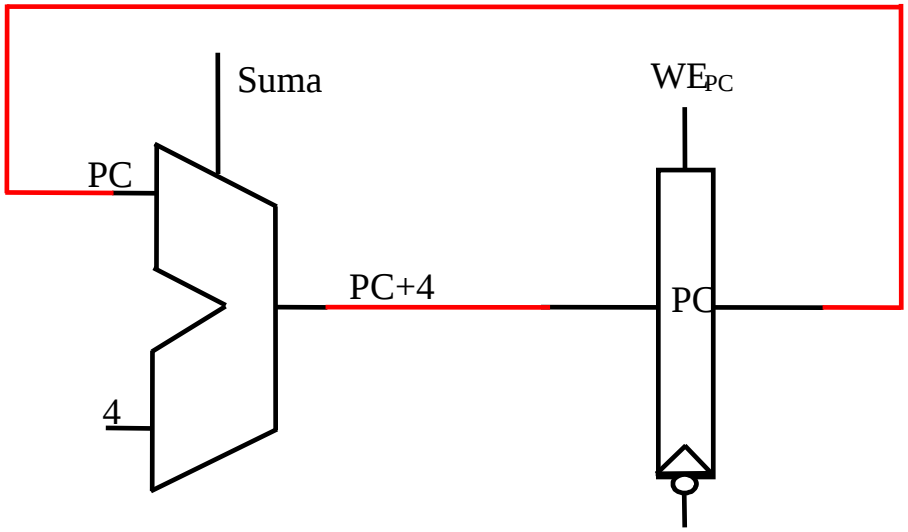
Procesador Monociclo

- Recursos Almacenamiento.
  - Registros. Contador de Programa



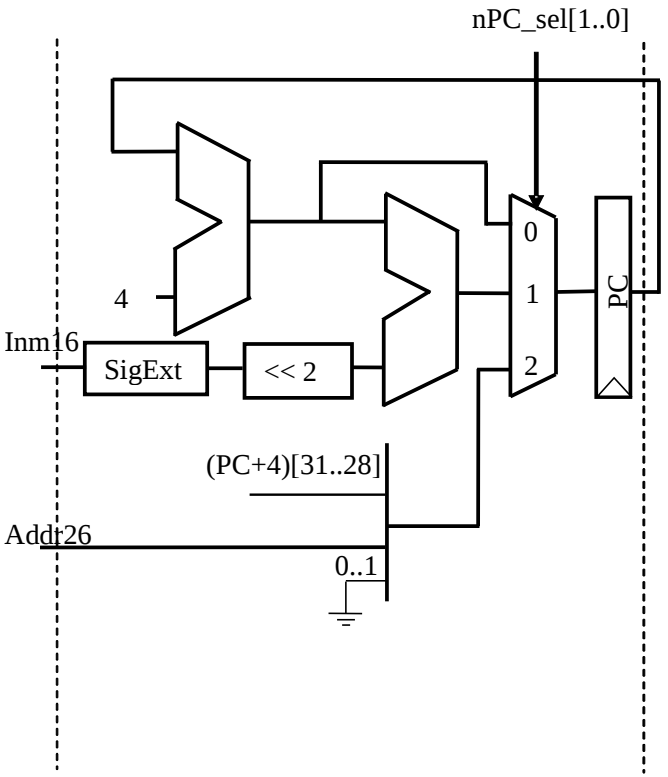
Procesador Monociclo

- Camino Datos. Determinación del próximo valor del PC.
  - PC = PC + 4



Procesador Monociclo

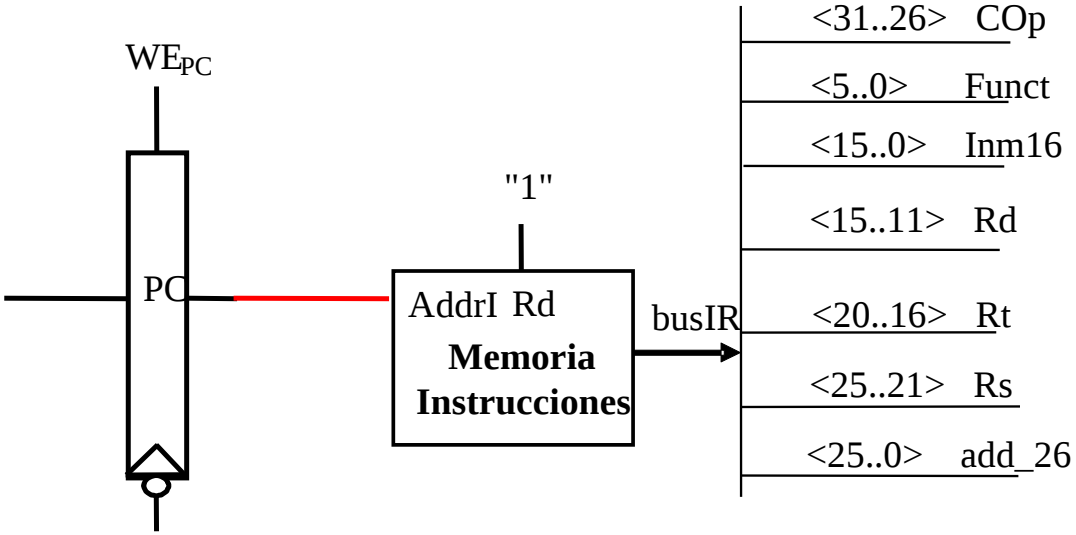
- Camino Datos. Determinación del próximo valor del PC.
  - Bifurcaciones y salto.



nPCsel	
00	PC = PC +4
01	PC = PC +4 + [Sign_ext(Inm16)]*4
10	PC = (PC +4)&0xF0000000+(addr_26*4)

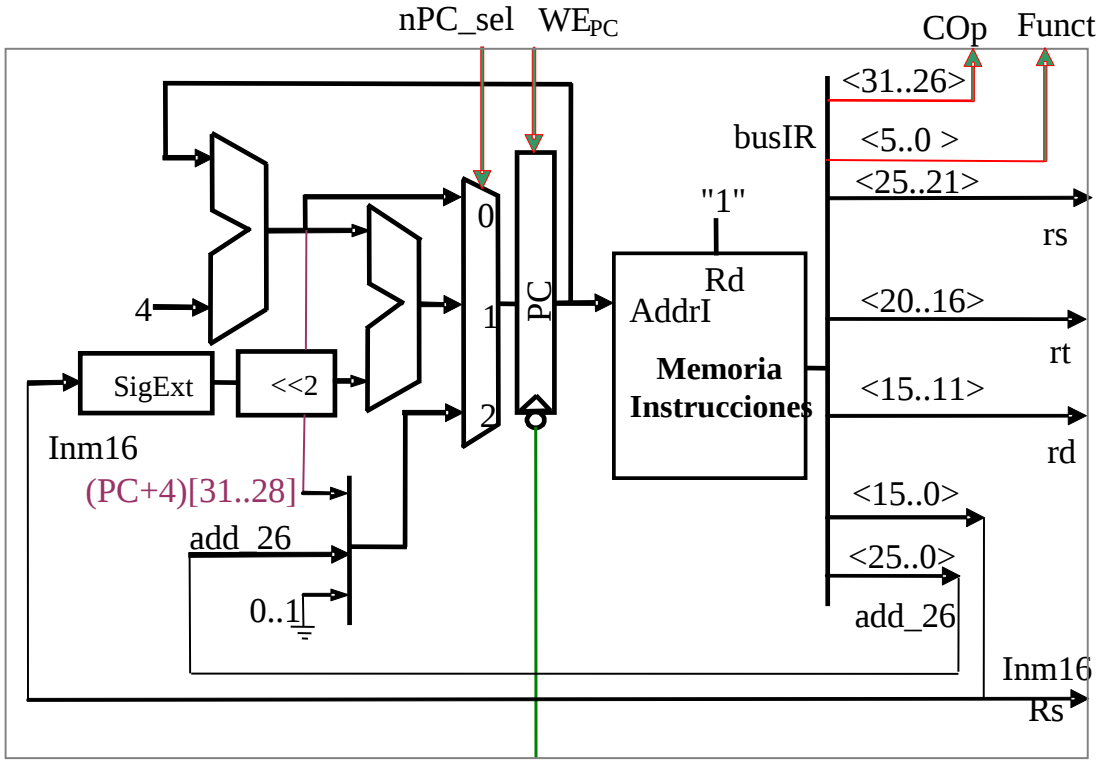
Procesador Monociclo

- Camino de datos entre PC y Memoria de Programa.



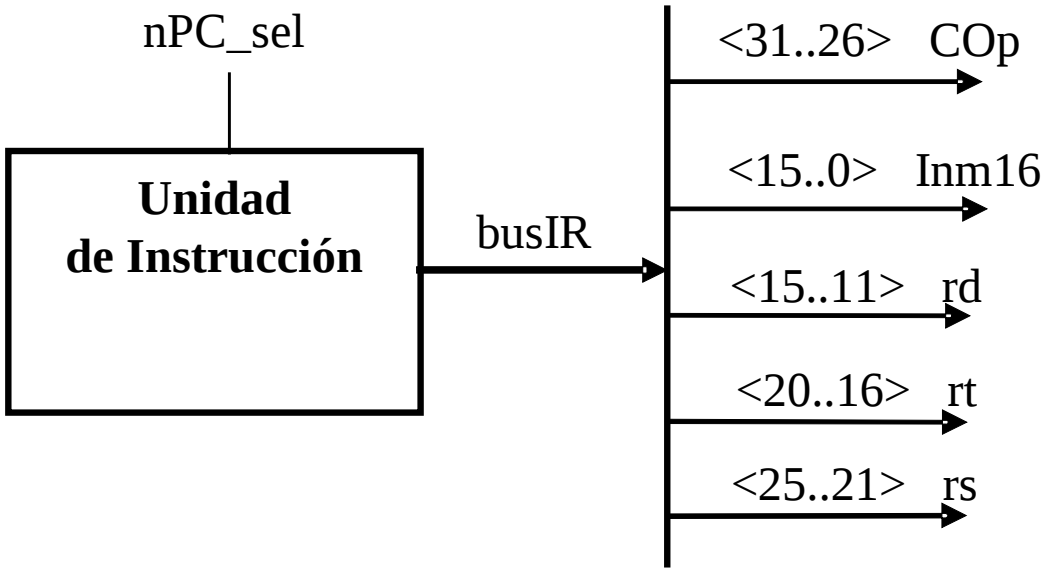
Procesador Monociclo

- Unidad que determina la próxima instrucción a ejecutar, con la memoria de instrucciones.



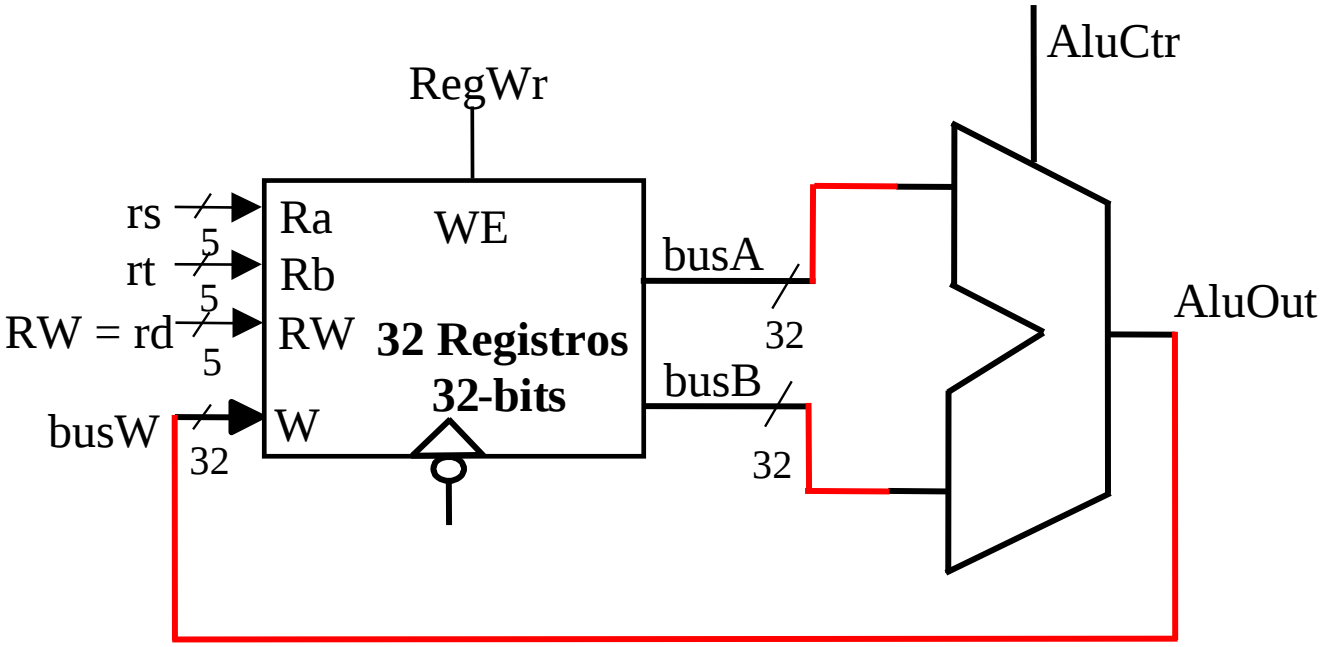
Procesador Monociclo

- Abstracción del diagrama anterior en un bloque que determina y decodifica la instrucción a ejecutar, y que además calcula la dirección de la próxima instrucción a realizar.



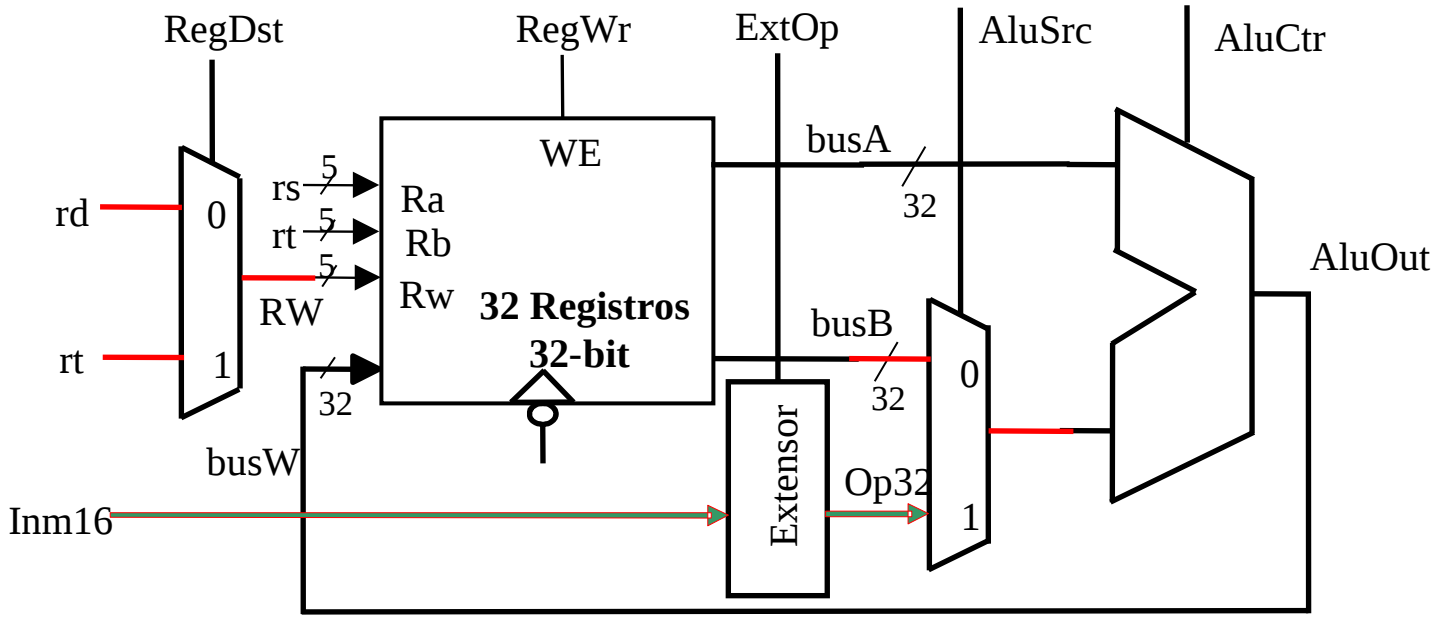
Procesador Monociclo

- Camino de Datos para operaciones de tipo R.
  - Arreglo de registros y la unidad aritmético lógica, para poder realizar las transferencias físicas que implementan las operaciones de suma y resta.



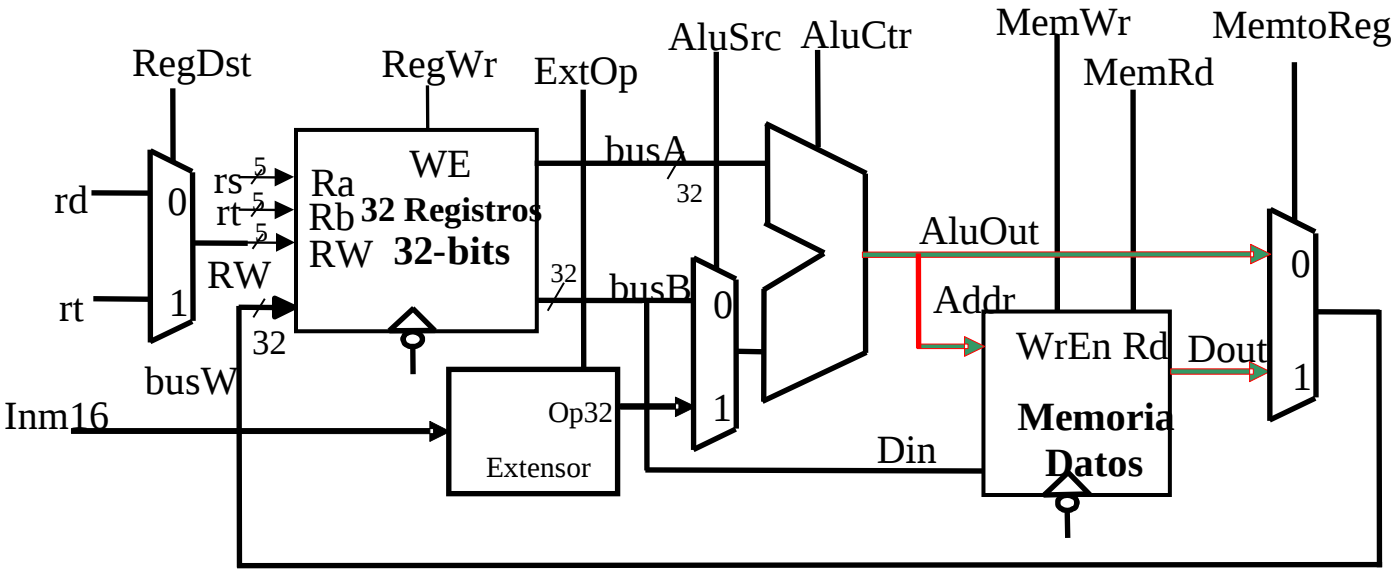
Procesador Monociclo

- Modificaciones al camino de datos para poder procesar instrucciones inmediatas.



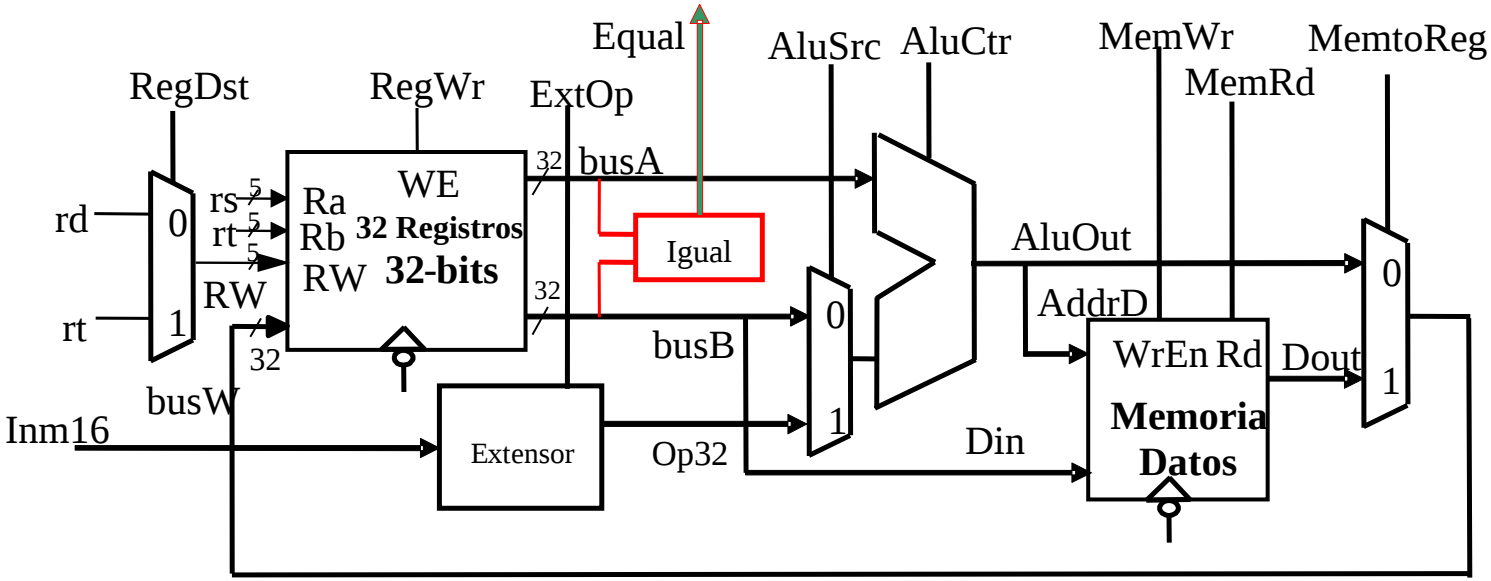
Procesador Monociclo

- Camino de datos para acceder la memoria de datos.



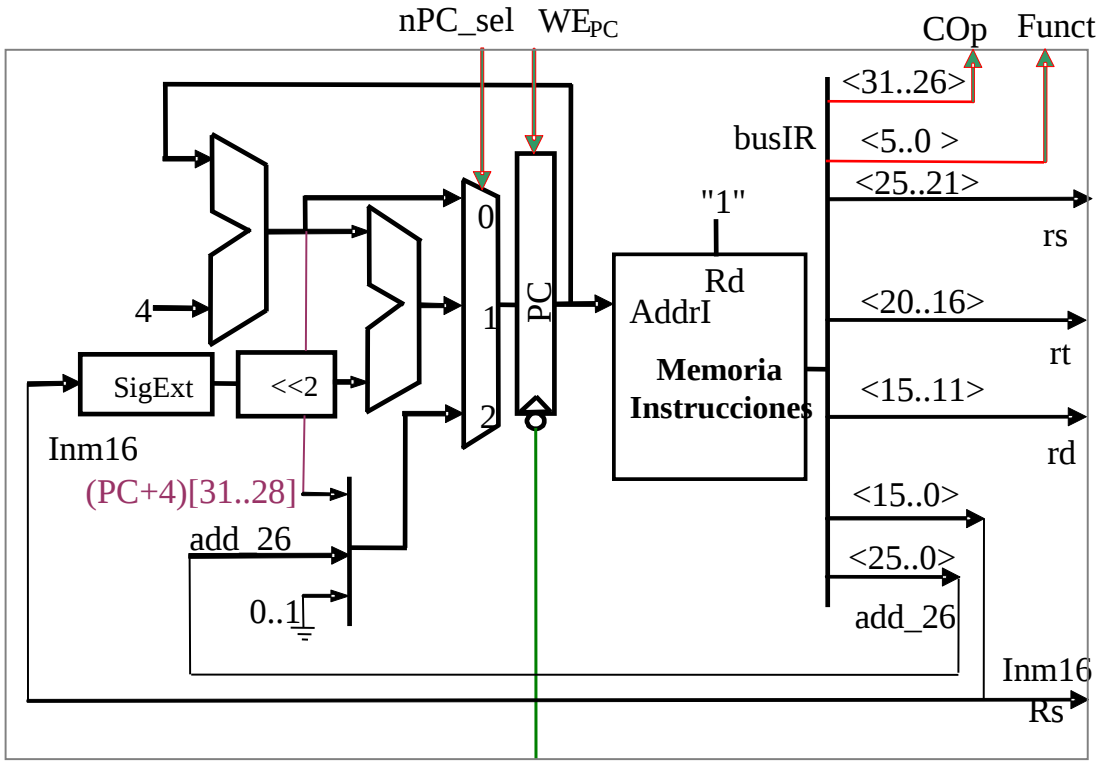
Procesador Monociclo

- Detector de igualdad de busA y busB



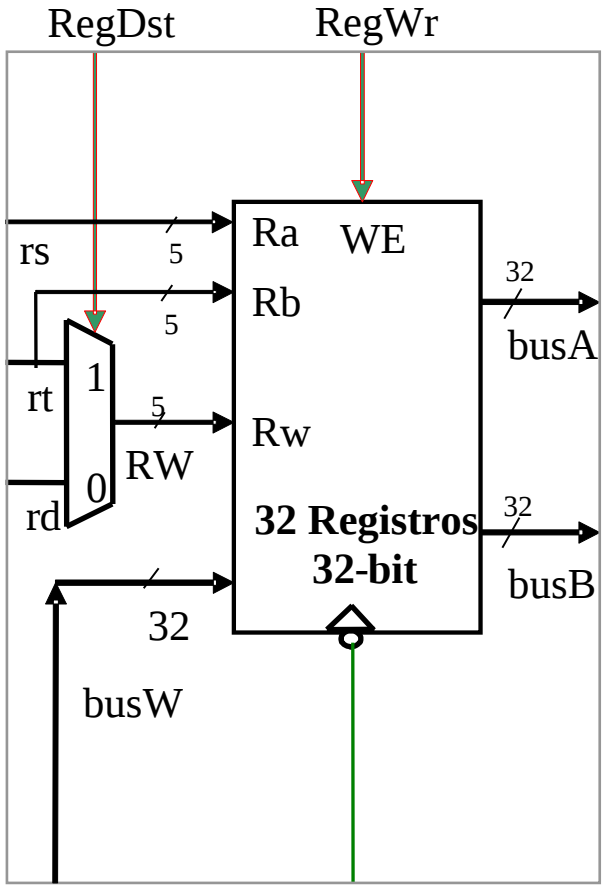
Procesador Monociclo

- Simplificación Considerando unidades
  - Unidad Instrucción



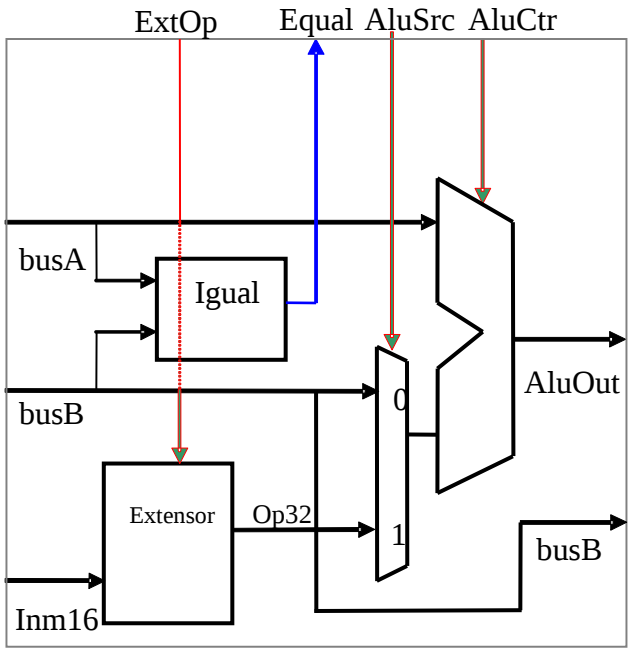
Procesador Monociclo

- Simplificación Considerando unidades
  - Unidad Registros



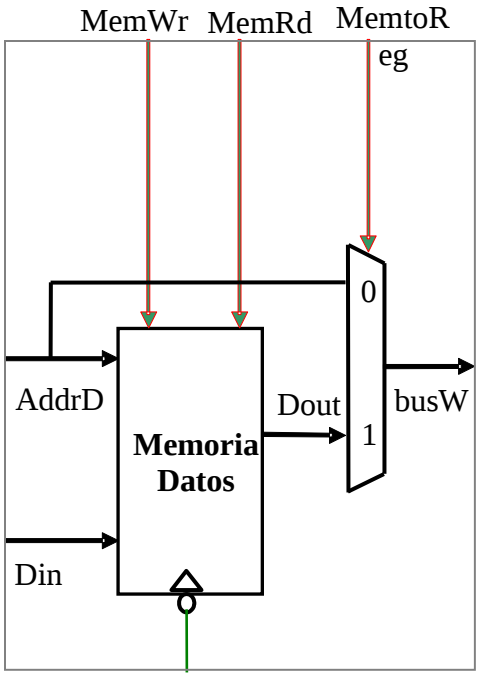
Procesador Monociclo

- Simplificación Considerando unidades
  - Unidad Operaciones



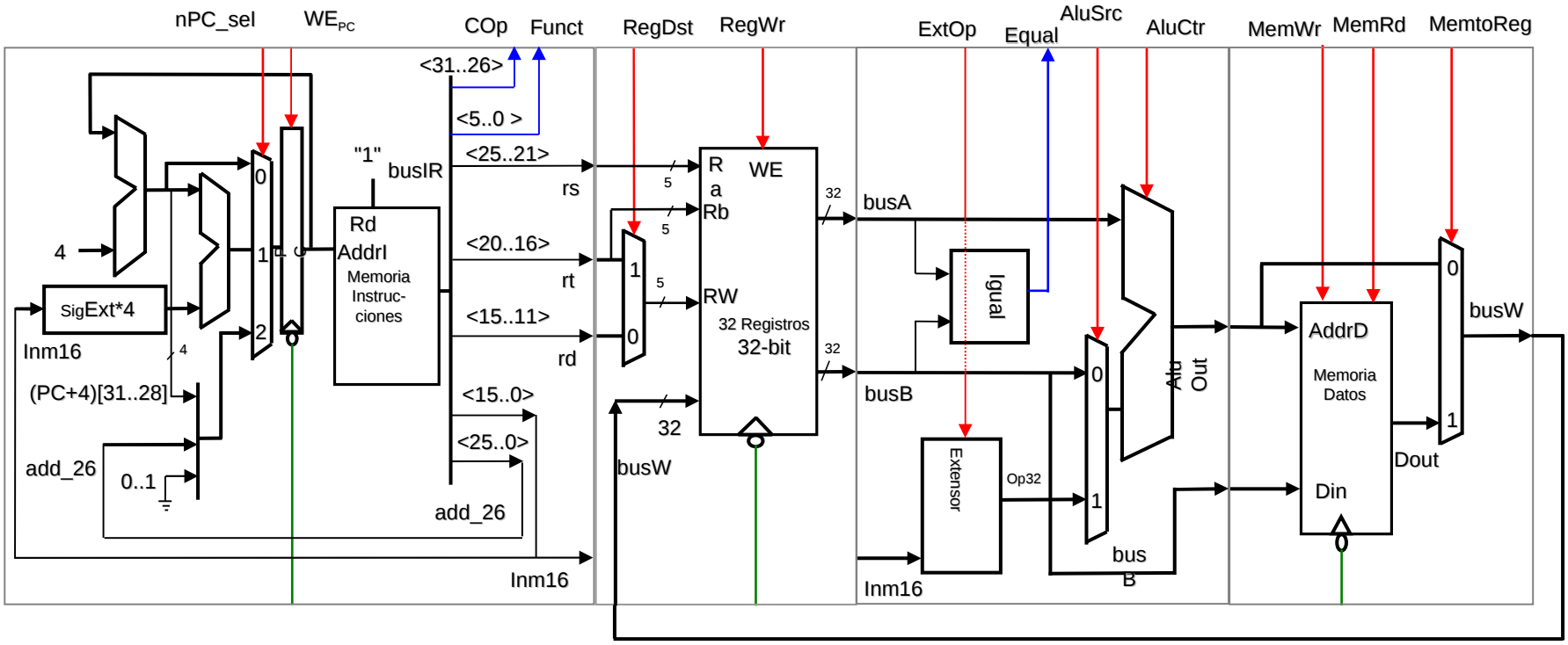
Procesador Monociclo

- Simplificación Considerando unidades
  - Unidad Memoria



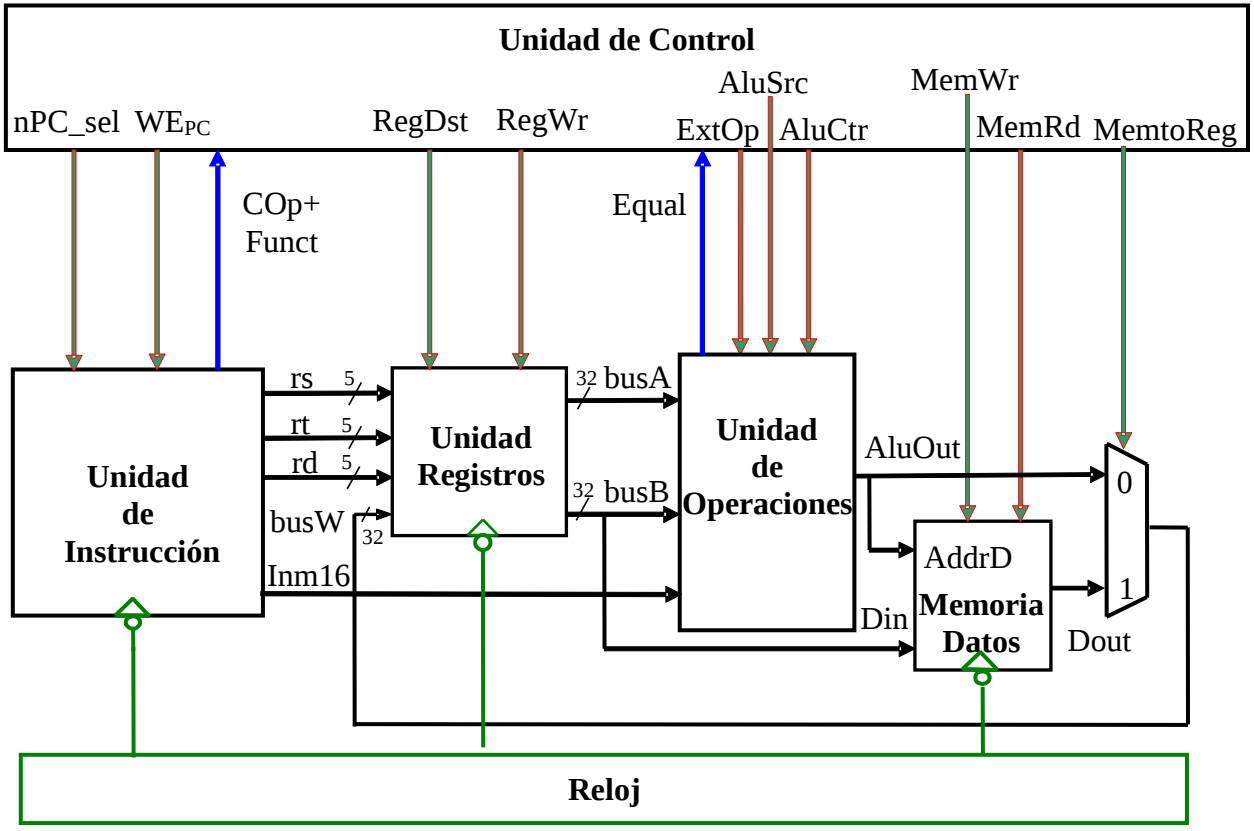
Procesador Monociclo

- Esquema General



Procesador Monociclo

- Esquema General por unidad



Procesador Monociclo

- Transferencias físicas de datos.
  - Cada transferencia lógica se implementa como un conjunto de transferencias físicas.
  - Transferencia física es la descripción de un movimiento de datos entre los recursos del camino de datos.
  - Muestra lo que sucede en el camino de datos.

Procesador Monociclo

- Transferencias físicas de datos.
  - Movimientos son debidos a:
    - Conexiones permanentes entre recursos
    - Señales de control.
  - Para cada instrucción:
    - Se hace referencia a través de un mnemónico.
    - Se anota la transferencia lógica.
    - Y luego el conjunto de transferencias físicas que la desarrollan.

Procesador Monociclo

- Transferencias físicas de datos.
  - Las transferencias físicas se agrupan de acuerdo a la unidad en que se realizan.
  - Se muestran por líneas, pero toda la electrónica que las representa está funcionando en paralelo, y la información fluye en serie a través de los recursos.

Procesador Monociclo

- Transferencias físicas de datos.
  - **ADD**: R[rd] ⇐ R[rs] + R[rt]; PC ⇐ PC + 4

AddrI=PC, busIR=MemInst[AddrI], PC=PC+4,	Unidad Instrucción.
Ra=rs, Rb=rt, RW=rd, busA=R[Ra], busB=R[Rb],	Unidad Registros. Lectura.
AluOut=add(busA, busB),	Unidad Operaciones.
busW=AluOut, R[RW]=busW.	Unidad Registros. Escritura.

  - **SUB**: R[rd] ⇐ R[rs] – R[rt]; PC ⇐ PC + 4

AddrI=PC, busIR=MemInst[AddrI], PC=PC+4,	Unidad Instrucción.
Ra=rs, Rb=rt, RW=rd, busA=R[Ra], busB=R[Rb],	Unidad Registros. Lectura.
AluOut= sub(busA, busB),	Unidad Operaciones.
busW=AluOut, R[RW]=busW.	Unidad Registros. Escritura.

Procesador Monociclo

- Transferencias físicas de datos.
  - **SLT**: R[rd] ⇐ R[rs] < R[rt] ? 1: 0 ; PC ⇐ PC + 4

AddrI=PC, busIR=MemInst[AddrI], PC=PC+4, Unidad Instrucción.
Ra=rs, Rb=rt, RW=rd, busA=R[Ra], busB=R[Rb], Unidad Registros. Lectura.
AluOut=slt(busA, busB), Unidad Operaciones.
busW=AluOut, R[RW]=busW. Unidad Registros. Escritura.

  - **ORI**: R[rt] ⇐ R[rs] | zero_ext(Inm16); PC ⇐ PC + 4

AddrI=PC, busIR=MemInst[AddrI], PC=PC+4, Unidad Instrucción.
Ra=rs, RW=rt, busA=R[Ra], Unidad Registros. Lectura.
Op32 = zero_ext(Inm16), AluOut=or(busA, Op32), Unidad Operaciones.
busW=AluOut, R[RW]=busW. Unidad Registros. Escritura.

Procesador Monociclo

- Transferencias físicas de datos.
  - **LOAD**: R[rt] ⇐ MEM[R[rs] + sign_ext(Inm16)]; PC ⇐ PC + 4

AddrI=PC, busIR=MemInst[AddrI], PC=PC+4,	Unidad Instrucción.
Ra=rs, RW=rt, busA=R[Ra],	Unidad Registros. Lectura.
Op32 = signext(Inm16), AluOut=add(busA , Op32),	Unidad Operaciones.
AddrD=AluOut, Dout = MemDat[AddrD],	Unidad Memoria Datos.
busW=Dout, R[RW]=busW.	Unidad Registros. Escritura.

  - **STORE**: MEM[R[rs] + sign_ext(Inm16)] ⇐ R[rt]; PC ⇐ PC + 4

AddrI=PC, busIR=MemInst[AddrI], PC=PC+4,	Unidad Instrucción.
Ra=rs, Rb=rt, busA=R[Ra], busB=R[Rb],	Unidad Registros. Lectura.
Op32 = signext(Inm16), AluOut= add(busA , Op32),	Unidad Operaciones.
AddrD=AluOut, Din = busB, MemDat[AddrD]=Din.	Unidad Memoria Datos.

Procesador Monociclo

- Transferencias físicas de datos.
  - **BEQ**: if (R[rs] == R[rt]) PC ⇐(PC +4)+ sign_ext(Inm16)] *4 else PC ⇐PC + 4

AddrI=PC, busIR=MemInst[AddrI],	Unidad Instrucción.
Ra=rs, Rb=rt, busA=R[Ra], busB=R[Rb],	Unidad Registros. Lectura.
if (Equal)	Unidad Operaciones.
PC = (PC +4)+ sign_ext(Inm16)] *4 else PC=PC+4.	Unidad Instrucción.

  - **J UMP**: PC ⇐(PC +4)&0xF0000000+ add_26 *4

AddrI=PC, busIR=MemInst[AddrI],	Unidad Instrucción.
PC = (PC +4)&0xF0000000+ add_26 *4.	Unidad Instrucción

Procesador Monociclo

- Transferencias físicas de datos.
  - Valores Conceptuales que toman las Señales de Control

nPC_sel: "+4","Branch","Jump"	Unidad Instrucción.
WEPC: "1" Escribe en registro PC.	Unidad Instrucción.
RegDst: “rt”, “rd”	Unidad Registros. Lectura.
ExtOp: “zero”, “sign”	Unidad Operaciones.
AluSrc: "busB","Op32"	Unidad Operaciones.
AluCtr: “add”, “sub”, “or”, “slt”	Unidad Operaciones.
MemWr: "1" Escribe en la memoria.	Unidad Memoria Datos.
MemRd: "1" Lee desde la memoria de datos.	Unidad Memoria Datos.
MemtoReg:"alu","mem"	Unidad Registros. Escritura.
RegWr: "1" escribe busW en el registro especificado en RW.	Unidad Registros. Escritura.

Procesador Monociclo

- Valores de las señales de control para activar las transferencias lógicas.
  - Valores que toman las señales de control para desarrollar las diferentes transferencias lógicas necesarias para cada instrucción.

Procesador Monociclo

- Valores de las señales de control para activar las transferencias lógicas.
  - **ADD:** R[rd] ⇐ R[rs] + R[rt]; PC ⇐ PC + 4

nPC_sel = “+4”, WEPC =1,	Unidad Instrucción.
RegDst = "rd",	Unidad Registros. Lectura.
AluSrc = "busB", AluCtr = “add”, ExtOp=∅,	Unidad Operaciones.
MemWr=0, MemRd=∅,	Unidad Memoria Datos.
Memtoreg="alu", RegWr=1.	Unidad Registros. Escritura.

  - **SUB:** R[rd] ⇐ R[rs] – R[rt]; PC ⇐ PC + 4

nPC_sel = “+4”, WEPC=1 ,	Unidad Instrucción.
RegDst = "rd",	Unidad Registros. Lectura.
AluSrc = "busB", AluCtr = “sub”, ExtOp=∅,	Unidad Operaciones.
MemWr=0, MemRd=∅,	Unidad Memoria Datos.
Memtoreg="alu", RegWr=1.	Unidad Registros. Escritura.

Procesador Monociclo

- Valores de las señales de control para activar las transferencias lógicas.
  - **SLT**: R[rd] ⇐ R[rs] < R[rt] ? 1: 0 ; PC ⇐ PC + 4

nPC_sel = “+4”, WEPC=1 ,	Unidad Instrucción.
RegDst = "rd",	Unidad Registros. Lectura.
AluSrc = "busB", AluCtr = “slt”, ExtOp=∅,	Unidad Operaciones.
MemWr=0, MemRd=∅,	Unidad Memoria Datos.
Memtoreg="alu", RegWr=1.	Unidad Registros. Escritura.

  - **ORI:** R[rt] ⇐ R[rs] + zero_ext(Inm16); PC ⇐ PC + 4

nPC_sel = “+4”, WEPC =1,	Unidad Instrucción.
RegDst = "rt",	Unidad Registros. Lectura.
AluSrc = "Op32", ExtOp = “zero”, AluCtr = “or”,	Unidad Operaciones.
MemWr=0, MemRd=∅,	Unidad Memoria Datos.
Memtoreg="alu", RegWr=1.	Unidad Registros. Escritura.

Procesador Monociclo

- Valores de las señales de control para activar las transferencias lógicas.
  - **LOAD**: R[rt] ⇐MEM[R[rs] + sign_ext(Inm16)]; PC ⇐PC + 4

nPC_sel = “+4”, WEPC=1 ,	Unidad Instrucción.
RegDst = "rt",	Unidad Registros. Lectura.
AluSrc = "Op32", ExtOp = “sign”, AluCtr = “add”,	Unidad Operaciones.
MemRd=1, MemWr=0,	Unidad Memoria Datos.
Memtoreg="mem", RegWr=1.	Unidad Registros. Escritura.

  - **STORE**: MEM[R[rs] + sign_ext(Inm16)] ⇐R[rt]; PC ⇐PC + 4

nPC_sel = “+4”, WEPC =1,	Unidad Instrucción.
RegDst = ∅,	Unidad Registros. Lectura.
AluSrc = "Op32", ExtOp = “sign”, AluCtr = “add”,	Unidad Operaciones.
MemWr=1, MemRd=0,	Unidad Memoria Datos.
Memtoreg=∅, RegWr=0.	Unidad Registros. Escritura

Procesador Monociclo

- Valores de las señales de control para activar las transferencias lógicas.
  - **BEQ**: if (R[rs] == R[rt]) PC ⇐(PC +4) + sign_ext(Inm16)] *4; else PC ⇐PC + 4

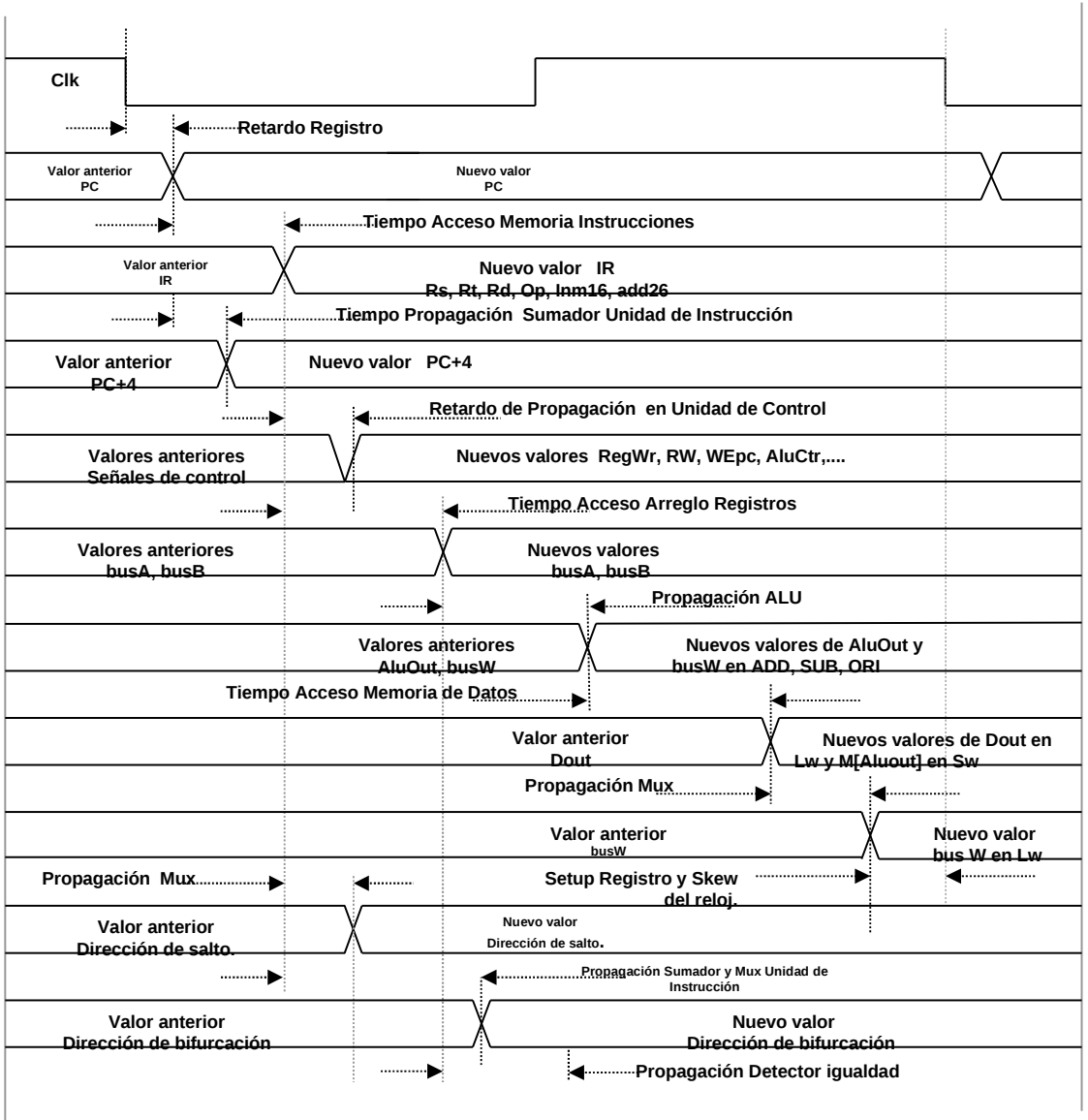
if (Equal) nPC_sel="Br"; else nPC_sel="+4"; WEPC =1.	Unidad Instrucción.
RegDst = ∅,	Unidad Registros. Lectura.
AluSrc = ∅, ExtOp = ∅ , AluCtr = ∅ ,	Unidad Operaciones.
MemWr=0, MemRd=∅,	Unidad Memoria Datos.
Memtoreg=∅, RegWr=0.	Unidad Registros. Escritura

  - **J:** PC ⇐(PC +4)&0xF0000000+ add_26 *4

nPC_sel = "Jmp", WEPC=1 .	Unidad Instrucción.
RegDst = ∅,	Unidad Registros. Lectura.
AluSrc = ∅, ExtOp = ∅ , AluCtr = ∅ ,	Unidad Operaciones.
MemWr=0, MemRd=∅.	Unidad Memoria Datos.
Memtoreg=∅, RegWr=0.	Unidad Registros. Escritura

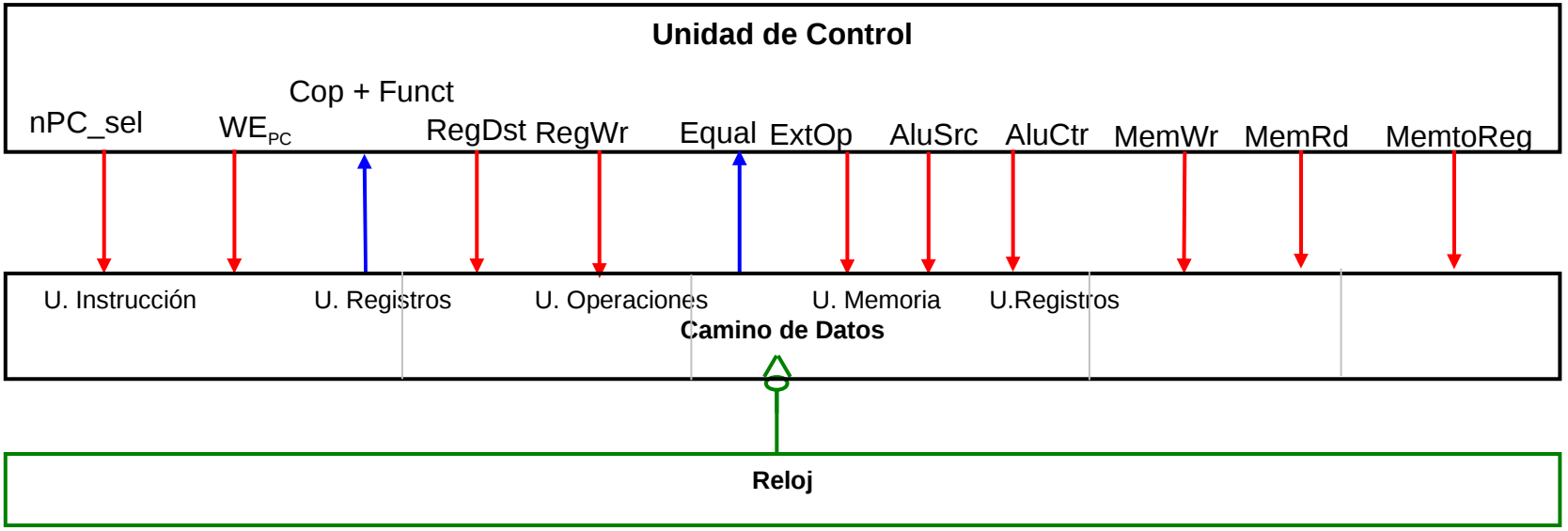
Procesador Monociclo

- Diagrama de tiempos para todas las instruccion es del procesador



Procesador Monociclo

- **Diseño de la Unidad de Control.**
  - En el modelo monociclo la unidad de control es una red combinacional que tiene como entradas el código de operación-funct y la condición Equal; y como salidas las señales de Control.



Procesador Monociclo

- **Diseño de la Unidad de Control.**
  - **Tabla de verdad de la Unidad de Control Monociclo**

Op	**Bnegate**	**Operación**	
		Decimal	Binario
and	ø	0	00
or	ø	1	01
add	0	2	10
slt	1	3	11
sub	1	2	10

Procesador Monociclo

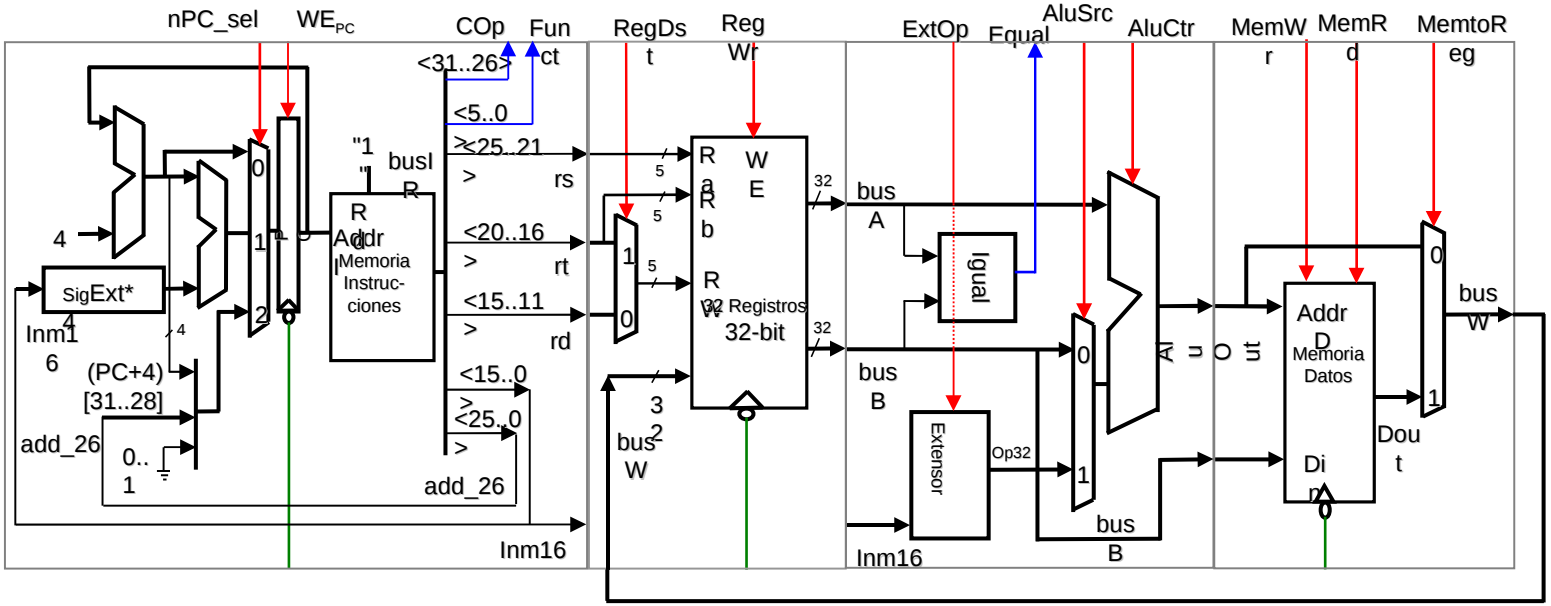
- **Diseño de la Unidad de Control.**
  - **Tabla de verdad de la Unidad de Control Monociclo**

OP	Funct		Nemo.
000000	100000	R	**add**
000000	100010	R	**sub**
000000	101010	R	**slt**
001101	ØØØØØØ	I	**ori**
100011	ØØØØØØ	I	**lw**
101011	ØØØØØØ	I	**sw**
000100	ØØØØØØ	I	**beq**
000010	ØØØØØØ	J	**j**
000000	100100	R	**and**
000000	101010	R	**slt**

Procesador Monociclo

- **Diseño de la Unidad de Control.**
  - **Tabla de verdad de la Unidad de Control Monociclo**

Op+funct	Eq	WE_{PC}	nPC_sel	Ext Op	Alu Src	Alu Ctr	Reg Dst	Mem Wr	Mem Rd	Mem toReg	Reg Wr
Add	ø	1	00	ø	0	010	0	0	ø	0	1
Sub	ø	1	00	ø	0	110	0	0	ø	0	1
Slt	ø	1	00	ø	0	111	0	0	ø	0	1
Ori	ø	1	00	0	1	001	1	0	ø	0	1
Load	ø	1	00	1	1	010	1	0	1	1	1
Store	ø	1	00	1	1	010	ø	1	0	ø	0
Beq	1	1	01	ø	ø	øøø	ø	0	ø	ø	0
Beq	0	1	00	ø	ø	øøø	ø	0	ø	ø	0
Jmp	ø	1	10	ø	ø	øøø	ø	0	ø	ø	0
Entradas 13		**Salidas** 13									



Op+funct	Eq	WE_{PC}	nPC_sel	Ext Op	Alu Src	Alu Ctr	Reg Dst	Mem Wr	Mem Rd	Mem toReg	Reg Wr
Add	ø	1	00	ø	0	010	0	0	ø	0	1
Sub	ø	1	00	ø	0	110	0	0	ø	0	1
Slt	ø	1	00	ø	0	111	0	0	ø	0	1
Ori	ø	1	00	0	1	001	1	0	ø	0	1
Load	ø	1	00	1	1	010	1	0	1	1	1
Store	ø	1	00	1	1	010	ø	1	0	ø	0
Beq	1	1	01	ø	ø	øøø	ø	0	ø	ø	0
Beq	0	1	00	ø	ø	øøø	ø	0	ø	ø	0
Jmp	ø	1	10	ø	ø	øøø	ø	0	ø	ø	0
Entradas 13	**Salidas** 13										

Procesador Monociclo

- **Diseño de la Unidad de Control.**
  - **Tabla Resumen Unidad de Control Monociclo**

OP+Funct+Eq	Control[12..0]	Obs.
000000100000ø	100ø001000ø01	**Add**
000000100010ø	100ø011000ø01	**Sub**
000000101010ø	100ø111000ø01	**Slt**
001101øøøøøøø	1000100110ø01	**Ori**
100011øøøøøøø	1001101010111	**Load**
101011øøøøøøø	10011010ø10ø0	**Store**
000100øøøøøø1	101øøøøøø0øø0	**Beq**
000100øøøøøø0	100øøøøøø0øø0	**Beq**
000010øøøøøøø	110øøøøøø0øø0	**Jmp**

Procesador Monociclo

- **Diseño de la Unidad de Control.**
  - **La implementación de la unidad de control puede ser utilizando.**
    - **En base a compuertas lógicas (función mínima).**
    - **Dispositivos lógicos programables (por ejemplo: GAL).**
    - **O mediante EPROM.**