

## (66.20) Organizacion de Computadoras: TP 1

Christian Angelone (93971)  
christiangellone@gmail.com

Agustin Gaillard (94849)  
agufiuba@gmail.com

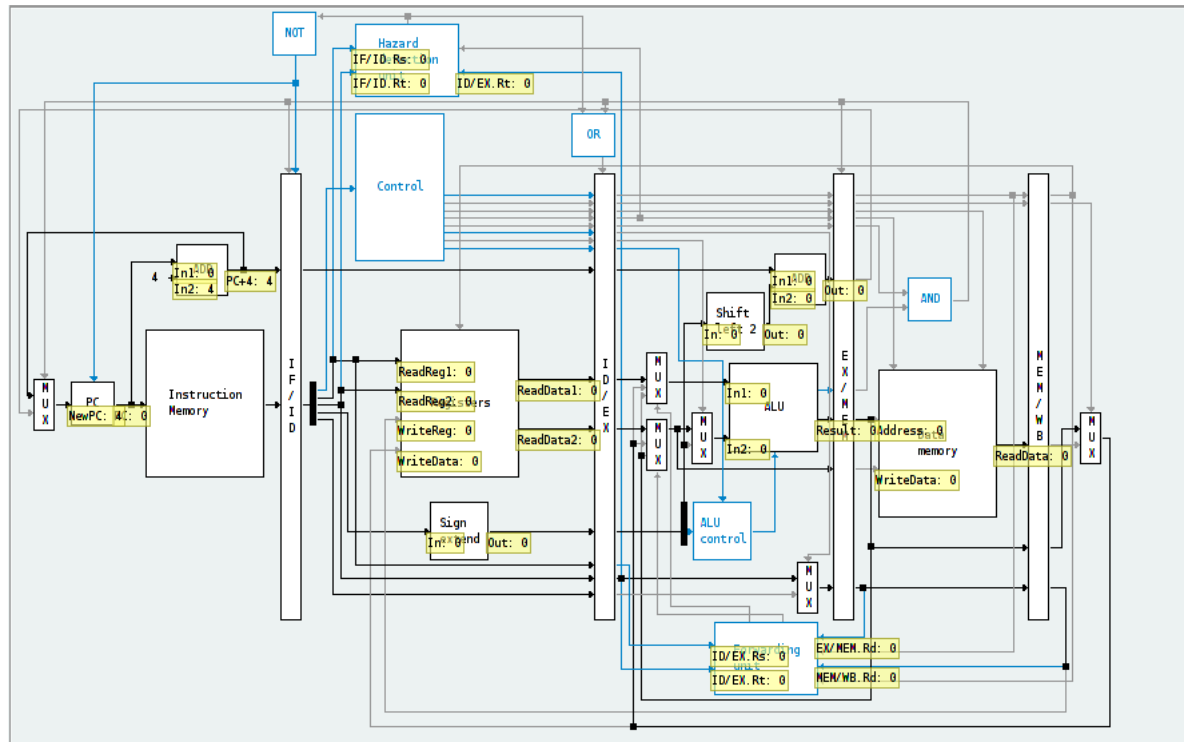
1er cuatrimestre 2019



### 1 Introduccion

Este trabajo practico, trata de mostrar distintos datapath y la implementacion de algunas instrucciones sobre ellos. repo: [https://github.com/christiangellone/tp3\\_datapath](https://github.com/christiangellone/tp3_datapath)

## 2 Diseño y Implementacion de "j" en un data-path multiciclo



Se aprovecho la implementacion en datapath de beq, para generar el jump incondicional

```
"j": {
  "type": "I",
  "args": ["offset"],
  "fields": {"op": 4, "rs": 0, "rt": 0, "imm": "#1", }
}
```

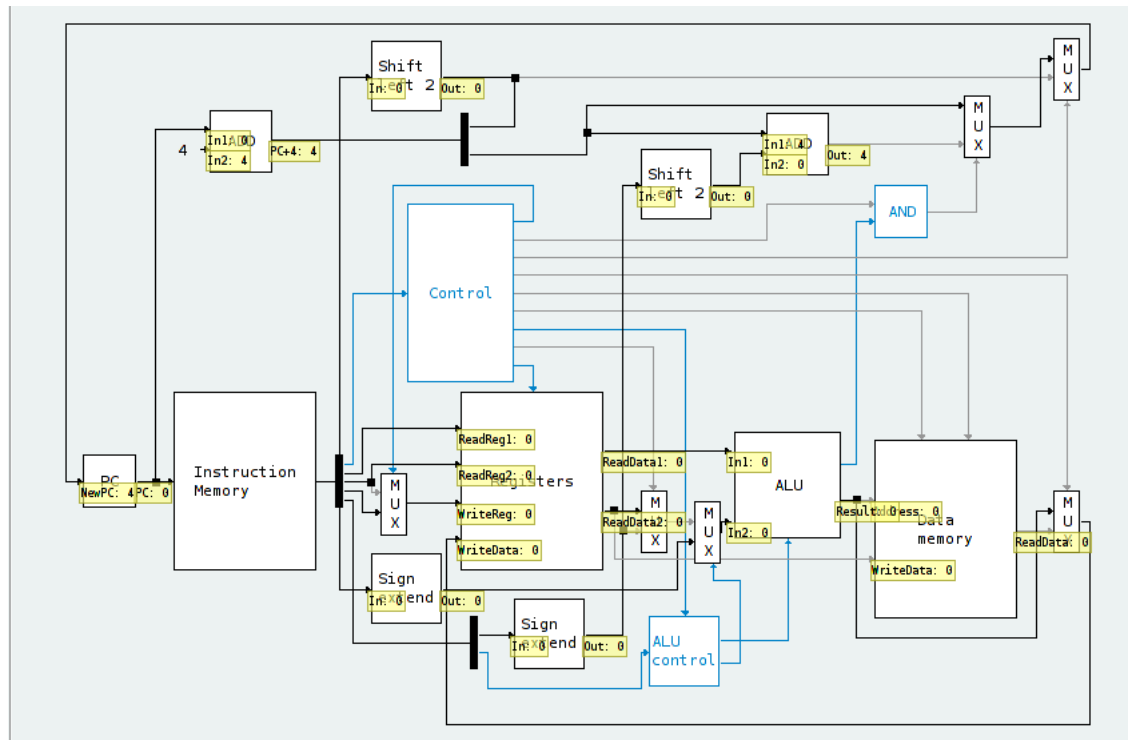
## 3 Ejecucion de casos de prueba y resultados

```
li $a0, 1
j jump
li $a0, 2
jump: li $a0, 3
```

Luego de ejecutar este codigo, a0 toma el valor 3, sin que la instruccion

siguiente al jump se ejecute.

## 4 Diseño y Implementacion de "sll" en un datapath uniciclo



Se utilizo un multiplexor para selecciona en ALU la opcion sll, y asi operar sobre el registro source.

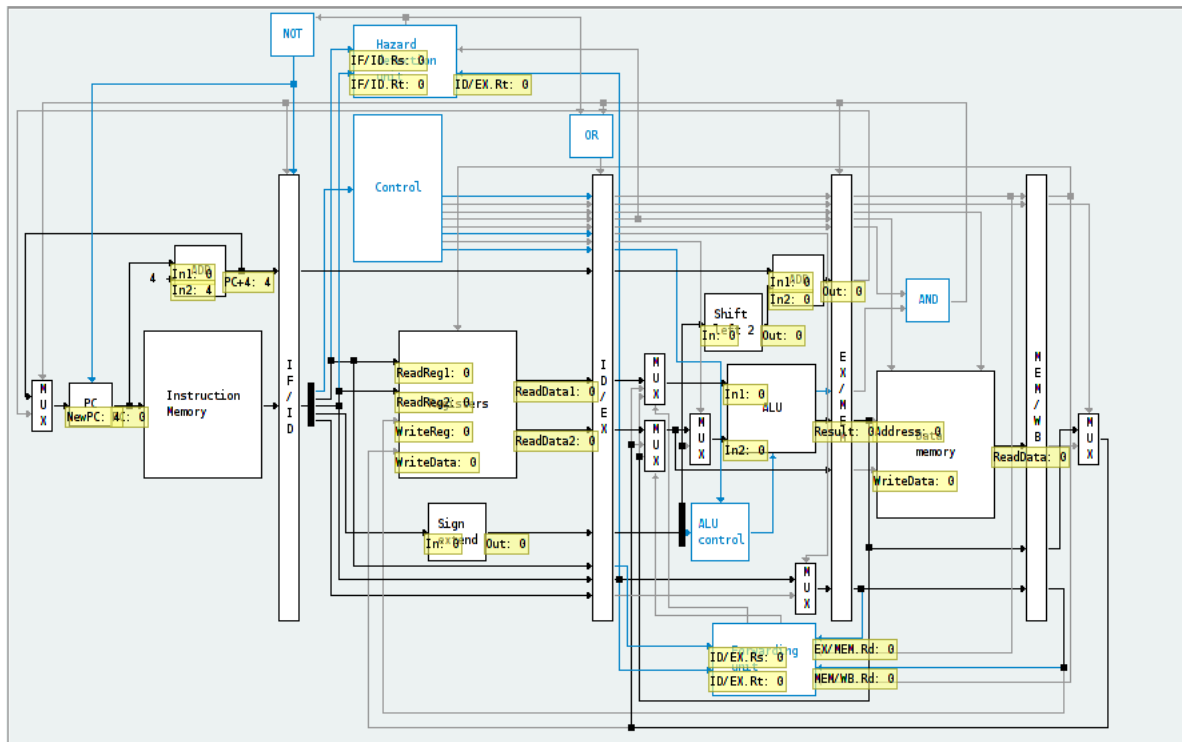
```
"sll": {
  "type": "R",
  "args": ["reg", "reg", "int"],
  "fields": {
    "op": 0,
    "rs": "#2",
    "rt": 0,
    "rd": "#1",
    "shamt": "#3",
    "func": 0
  },
  "desc": "$t1 = $t2 << $t3"
}
```

## 5 Ejecucion de casos de prueba y resultados

```
li $a0, 4
sll $a0, $a0, 2 # a0 = 16
```

Luego de ejecutar este codigo, a0 es multiplicado por 4

## 6 Diseno y Implementacion de "srl" en un datapath multiciclo



Se utilizo un multiplexor para selecciona en ALU la opcion srl, y asi operar sobre el registro source.

```
"srl": {
  "type": "R",
  "args": ["reg", "reg", "int"],
  "fields": {
    "op": 0,
    "rs": "#2",
    "rt": 0,
    "rd": "#1",
    "shamt": "#3",
    "func": 0
  }
}
```

```

},
"desc": "$t1 = $t2 >> $t3"
}

```

## 7 Ejecucion de casos de prueba y resultados

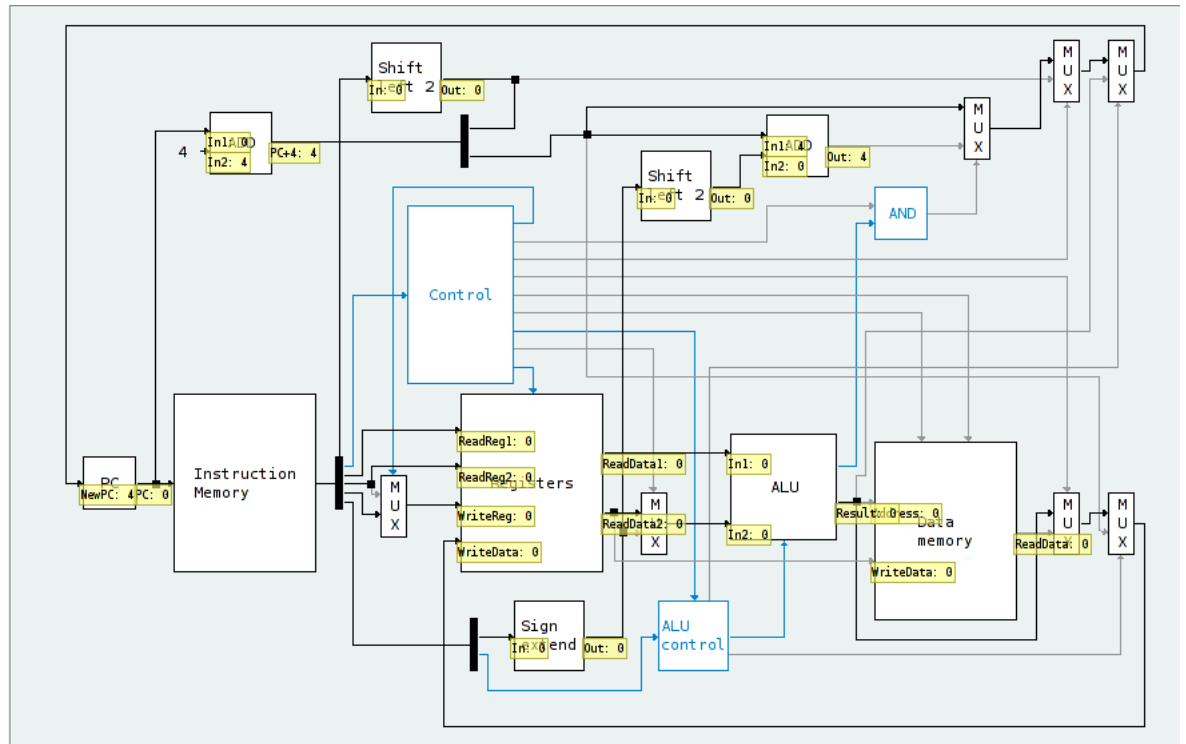
```

li $a0, 4
srl $a0, $a0, 2 # a0 = 1

```

Luego de ejecutar este codigo, a0 es dividido por 4

## 8 Diseno y Implementacion de "jalr" en un datapath uniclo



se asigno al registro ra el pc + 4 , y se copio al pc el registro source.

```

"jalr": {
  "type": "R",
  "args": ["reg"],
  "fields": {
    "op": 0,
    "rs": "#1",

```

```

        "rt": 0, "rd": 31,
        "shamt": 0,
        "func": 9
    },
    "desc": "$ra = PC+4; PC = $r"
}

```

## 9 Ejecucion de casos de prueba y resultados

```

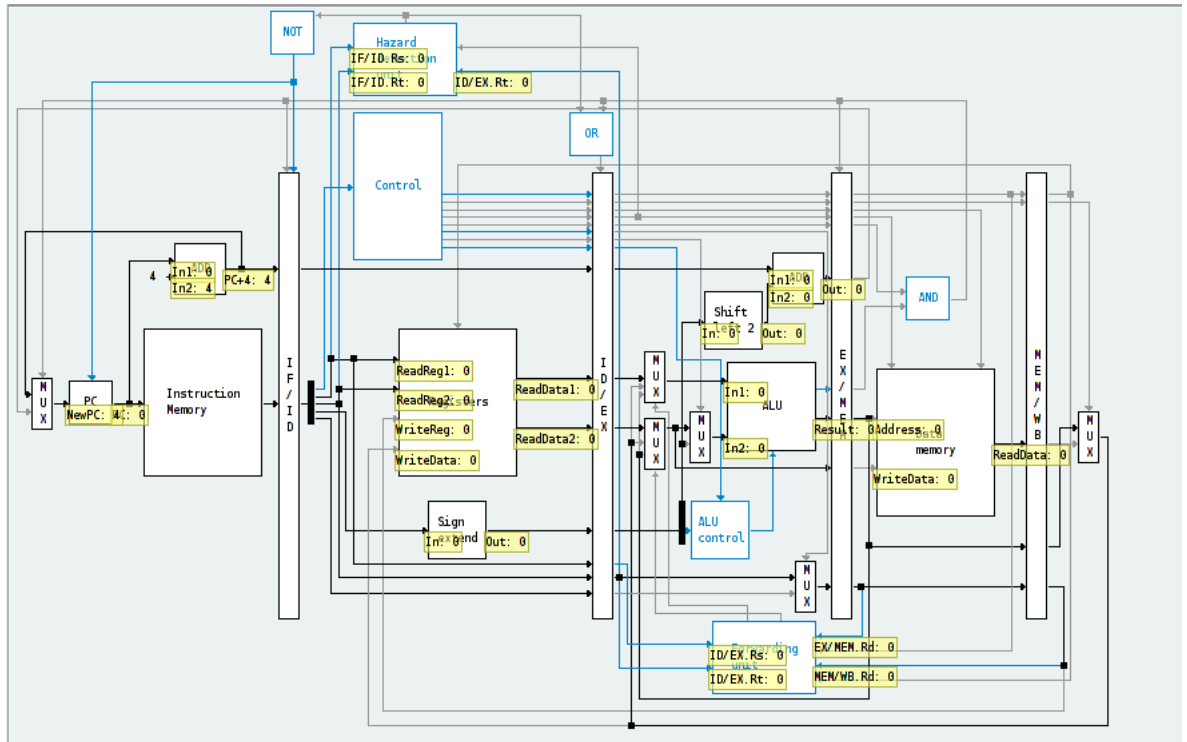
li $a0, jump
jalr $a0
li $a1, 2
nop
nop
nop
...
jump: li $a1, 3
jr $ra

```

Luego de ejecutar este codigo, jump es llamado sin que la siguiente instruc-

cion sea ejecutada hasta que se retorna de jump.

## 10 Diseno y Implementacion de "jalr" en un datapath multiciclo



se asigno al registro ra el pc + 4 , y se copio al pc el registro source.

```
"jalr": {
  "type": "R",
  "args": ["reg"],
  "fields": {
    "op": 0,
    "rs": "#1",
    "rt": 0, "rd": 31,
    "shamt": 0,
    "func": 9
  },
  "desc": "$ra = PC+4; PC = $r"
}
```

## 11 Ejecucion de casos de prueba y resultados

```

li $a0, jump
jalr $a0
li $a1, 2
nop
nop
nop
...
jump: li $a1, 3
jr $ra

```

Luego de ejecutar este código, jump es llamado sin que la siguiente instrucción sea ejecutada hasta que se retorna de jump

## 12 Conclusiones

- Adaptar datapaths multiciclo a nuevas instrucciones es más complejo que en un datapath uniciclo
- En general, la velocidad de ejecución de un programa en un datapath multiciclo es mayor a la de un datapath uniciclo
- El forwarding ayuda (cuando puede aplicarse) a la velocidad de ejecución de un programa. (datapaths multiciclo)