Universidade de Brasília
Departamento de Ciência da Computação

Aula 18
Exceções e Interrupções

Exceções e Interrupções

- Exceções são mudanças no fluxo de execução devido a eventos inesperados gerados internamente ao processador.
  - ecall : chamadas às rotinas do sistema
  - ebreak: chamada à rotina de Debug
  - instrução inválida
  - overflow em operações aritméticas (não são detectadas no RISC-V)
  - singularidades matemáticas da FPU (apenas sinalizadas pelo RISC-V)
  - ...

- Interrupções são mudanças no fluxo devido a eventos externos, tipicamente dispositivos de entrada e saída.
  - DMA (Direct Memory Access)
  - acesso ao barramento
  - solicitação de dispositivos
  - ...

Definições dependentes do fabricante: Intel usa o termo interrupção,
Ex.: int 0x21 : Dá acesso a 108 chamadas do sistema (similar ao a7 ecall)

Tratamento de Interrupções e Exceções

Para tratar a Interrupção/Exceção, Sistema Operacional necessita:

1) Conhecer o fato que gerou a Exceção/Interrupção

- Uso de Registrador de Causa (CAUSE)

  Um registrador especial é utilizado para codificar o motivo da Exceção/Interrupção.

- Uso de Interrupção Vetorizada

  Um vetor (endereço na memória) é utilizado para indicar os endereços para as rotinas de cada interrupção.

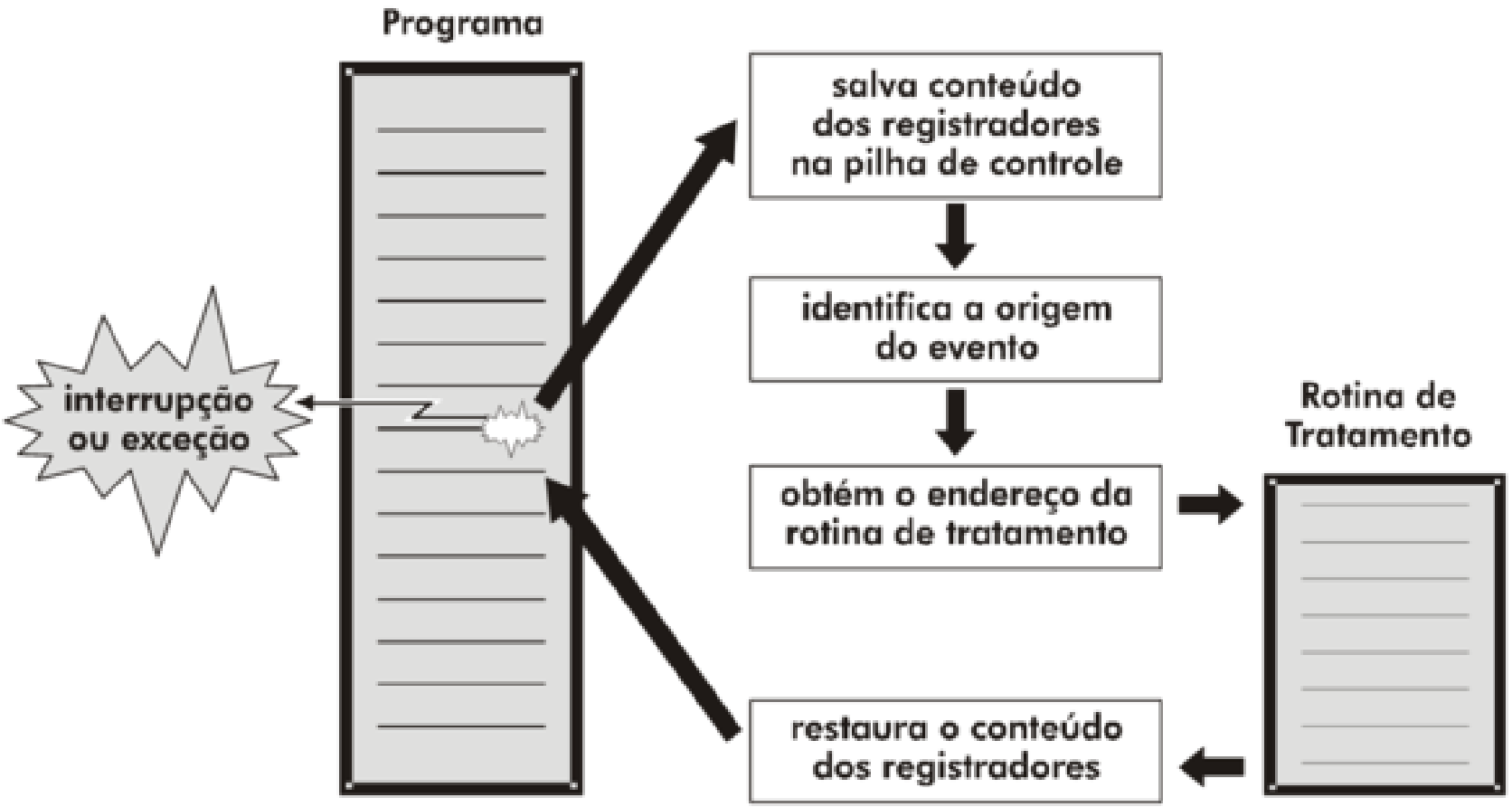
2) Conhecer o Endereço da instrução onde ocorreu a exceção/interrupção

- Registrador específico (EPC)

3) Executar uma rotina capaz de acessar recursos que podem não estar disponíveis ao usuário:

- ExceptionHandler

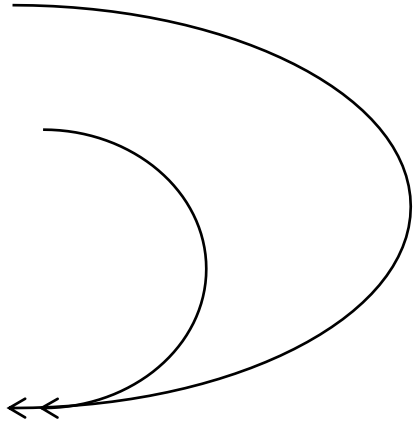
Tratamento da Exceção ou Interrupção



Tratamento de Interrupções e Exceções

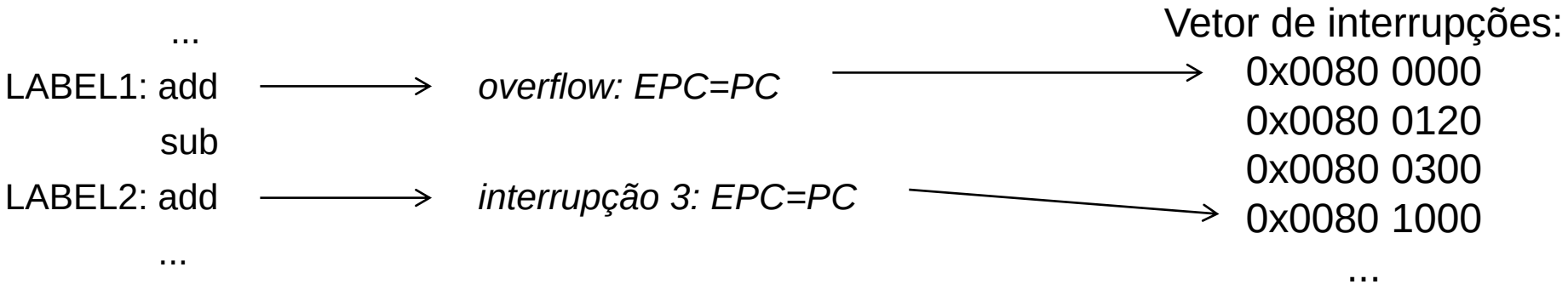
Uso do Registrador de Causa

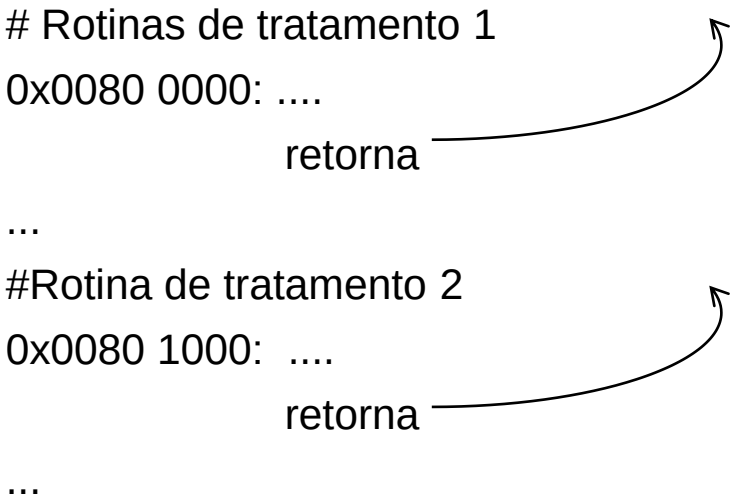
```
# Programa
            ...
LABEL1: add      ⟶    overflow: Cause=12 / EPC=PC
            sub
LABEL2: add      ⟶    interrupção 3: Cause=0 / EPC=PC
            ...

# Rotina de tratamento de exceção
ExceptionHandler: ....
                    se cause=0   então vá para xxxx
                    se cause=12 então vá para yyyy
                    ...
                xxxx:...
                    retorna
                yyyy:...
                    retorna
```



Tratamento de Interrupções e Exceções

Uso de Interrupção Vetorizada





Modos de Operação no RISC-V

- **Modo Usuário (*user mode*) :**

  Modo não privilegiado. Não pode acessar certos endereços da memória, algumas instruções não são permitidas. (Usuário com Sistema Operacional)

  Instruções: uret, ubreak

  Registradores: UCAUSE, USTATUS, UTVEC, UEPC, UIP, UIE, UTVAL, USCRATCH.

- **Modo Supervisor (*supervisor mode*):**

  Modo privilegiado. Pode acessar a toda memória e instruções específicas. (Sistema Operacional)

  Instruções: sret, sbreak

  Registradores: SCAUSE, SSTATUS, STVEC, SEPC, SIP, SIE, STVAL, SSCRATCH.

- **Modo Máquina (*machine mode*):**

  Sem nenhuma limitação de acesso (Usuário sem Sistema Operacional).

  Instruções: mret, mbreak, wfi

  Registradores: MCAUSE, MSTATUS, MTVEC, MEPC, MIP, MIE, MTVAL, MSCRATCH.

Exceções e Interrupções no RISC-V

- **Control and Status Registers (CSR)**
  - Espaço de 2^{12} = 4096 registradores dedicados às funções gerenciamento de exceções/interrupções, gerenciamento da memória, virtualização e outras atividades nos 3 modos de operação.

- No Rars há um sistema operacional mínimo logo usaremos o modo User.
- No Lab não há um sistema operacional, mas continuaremos usando o modo User para fins de compatibilidade com o Rars

CSR Address [11:10]	CSR Address [9:8]	CSR Address [7:6]	Hex	Use and Accessibility
User CSRs				
00	00	XX	0x000–0x0FF	Standard read/write
01	00	XX	0x400–0x4FF	Standard read/write
10	00	XX	0x800–0x8FF	Non-standard read/write
11	00	00–10	0xC00–0xCBF	Standard read-only
11	00	11	0xCC0–0xCFF	Non-standard read-only
Supervisor CSRs				
00	01	XX	0x100–0x1FF	Standard read/write
01	01	00–10	0x500–0x5BF	Standard read/write
01	01	11	0x5C0–0x5FF	Non-standard read/write
10	01	00–10	0x900–0x9BF	Standard read/write
10	01	11	0x9C0–0x9FF	Non-standard read/write
11	01	00–10	0xD00–0xDBF	Standard read-only
11	01	11	0xDC0–0xDFF	Non-standard read-only
Reserved CSRs				
XX	10	XX	Reserved	
Machine CSRs				
00	11	XX	0x300–0x3FF	Standard read/write
01	11	00–10	0x700–0x79F	Standard read/write
01	11	10	0x7A0–0x7AF	Standard read/write debug CSRs
01	11	10	0x7B0–0x7BF	Debug-mode-only CSRs
01	11	11	0x7C0–0x7FF	Non-standard read/write
10	11	00–10	0xB00–0xBBF	Standard read/write
10	11	11	0xBC0–0xBFF	Non-standard read/write
11	11	00–10	0xF00–0xFBF	Standard read-only
11	11	11	0xFC0–0xFFF	Non-standard read-only

Table 2.1: Allocation of RISC-V CSR address ranges.

Exceções e Interrupções no RISC-V

- **Control and Status Registers**
  Modo User

No Rars:

Control and Status

Registers | Floating Point

Name	Numb...	Value
ustatus	0	0x00000000
fflags	1	0x00000000
frm	2	0x00000000
fcsr	3	0x00000000
uie	4	0x00000000
utvec	5	0x00000000
uscratch	64	0x00000000
uepc	65	0x00000000
ucause	66	0x00000000
utval	67	0x00000000
uip	68	0x00000000
misa	769	0x40001128
cycle	3072	0x00000000
time	3073	0x00000000
instret	3074	0x00000000
cycleh	3200	0x00000000
timeh	3201	0x00000000
instreth	3202	0x00000000

Number	Privilege	Name	Description
		User Trap Setup	
0x000	URW	ustatus	User status register.
0x004	URW	uie	User interrupt-enable register.
0x005	URW	utvec	User trap handler base address.
		User Trap Handling	
0x040	URW	uscratch	Scratch register for user trap handlers.
0x041	URW	uepc	User exception program counter.
0x042	URW	ucause	User trap cause.
0x043	URW	utval	User bad address or instruction.
0x044	URW	uip	User interrupt pending.
		User Floating-Point CSRs	
0x001	URW	fflags	Floating-Point Accrued Exceptions.
0x002	URW	frm	Floating-Point Dynamic Rounding Mode.
0x003	URW	fcsr	Floating-Point Control and Status Register (frm + fflags).
		User Counter/Timers	
0xC00	URO	cycle	Cycle counter for RDCYCLE instruction.
0xC01	URO	time	Timer for RDTIME instruction.
0xC02	URO	instret	Instructions-retired counter for RDINSTRET instruction.
0xC03	URO	hpmcounter3	Performance-monitoring counter.
0xC04	URO	hpmcounter4	Performance-monitoring counter.
		⋮	
0xC1F	URO	hpmcounter31	Performance-monitoring counter.
0xC80	URO	cycleh	Upper 32 bits of cycle, RV32I only.
0xC81	URO	timeh	Upper 32 bits of time, RV32I only.
0xC82	URO	instreth	Upper 32 bits of instret, RV32I only.
0xC83	URO	hpmcounter3h	Upper 32 bits of hpmcounter3, RV32I only.
0xC84	URO	hpmcounter4h	Upper 32 bits of hpmcounter4, RV32I only.
		⋮	
0xC9F	URO	hpmcounter31h	Upper 32 bits of hpmcounter31, RV32I only.

Table 2.2: Currently allocated RISC-V user-level CSR addresses.

Exceções e Interrupções no RISC-V

- Instruções de controle e de acesso aos CSRs

Instrução	Descrição
`csrrc t0,csr,t1`	*CSR Read/Clear*: Lê o registrador CSR para t0 e “clear” (reseta) os bits do CSR de acordo com t1
`csrrci t0,csr,10`	*CSR Read/Clear Immediate*: Lê o registrador CSR para t0 e “clear” (reseta) os bits do CSR de acordo com o imediato
`csrrs t0,csr,t1`	*CSR Read/Set:* Lê o registrador CSR para t0 e seta os bits do CSR de acordo com t1
`csrrsi t0,csr,10`	*CSR Read/Set Immediate*: Lê o registrador CSR para t0 e seta os bits do CSR de acordo com o imediato
`csrrw t0,csr,t1`	*CSR Read/Write*: Lê o registrador CSR para t0 e copia o registrador t1 para o CSR
`csrrwi t0,csr,10`	*CSR Read/Write Immediate*: Lê o registrador CSR para t0 e copia o imediato para o CSR

Exceções e Interrupções no RISC-V

- **Codificação do Registrador de Causa**

Se bit ucause[31]=1 é uma interrupção

Se bit ucause[31]=0 é uma exceção

Interrupt	Exception Code	Description
1	0	User software interrupt
1	1	Supervisor software interrupt
1	2	*Reserved*
1	3	Machine software interrupt
1	4	User timer interrupt
1	5	Supervisor timer interrupt
1	6	*Reserved*
1	7	Machine timer interrupt
1	8	User external interrupt
1	9	Supervisor external interrupt
1	10	*Reserved*
1	11	Machine external interrupt
1	≥12	*Reserved*
0	0	Instruction address misaligned
0	1	Instruction access fault
0	2	Illegal instruction
0	3	Breakpoint
0	4	Load address misaligned
0	5	Load access fault
0	6	Store/AMO address misaligned
0	7	Store/AMO access fault
0	8	Environment call from U-mode
0	9	Environment call from S-mode
0	10	*Reserved*
0	11	Environment call from M-mode
0	12	Instruction page fault
0	13	Load page fault
0	14	*Reserved*
0	15	Store/AMO page fault
0	≥16	*Reserved*

Table 3.6: Machine cause register (`mcause`) values after trap.

Exceções e Interrupções no RISC-V

- Aceita tanto tratamento via registrador de causa quanto vetorizada

Se endereço em UTVEC/STVEC/MTVEC terminar em 00 trata-se de uma rotina única de tratamento de exceção localizada no endereço

BASE = { xTVEC[31:2], 00 }

Se endereço em UTVEC/STVEC/MTVEC terminar em 01 trata-se de tratamento vetorizado sendo o endereço calculado por:

Endereço = BASE + 4 $\times$ CAUSE

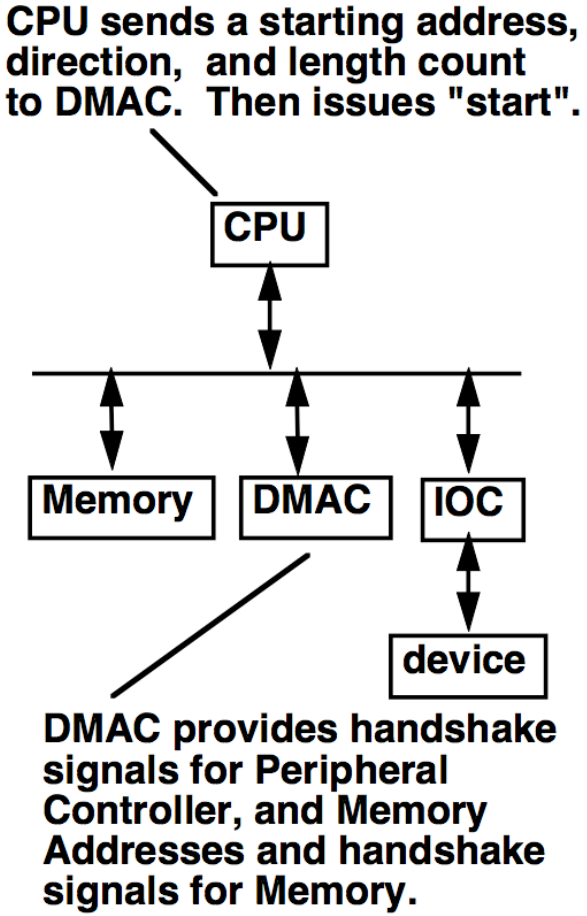
Operações de Entrada e Saída

- Por Polling: (software)
  - processador testa periodicamente se dispositivo está pronto para realizar a transferência de dados
  - Problema: toma muito tempo do processador

- Por Interrupção: (hardware)
  - o dispositivo avisa ao processador a sua disponibilidade
  - Problema: hardware mais complexo, processador deve suportar interrupções
    - antes de iniciar a execução da próxima instrução, processador verifica se existe uma solicitação de interrupção
    - caso haja, interrompe o processamento normal e executa uma rotina de tratamento de interrupções

Operações de Entrada e Saída

- Por DMA (Direct Memory Access)
  - dispositivo que gerencia a transferência de blocos de dados de forma independente da CPU
  - atua como mestre do barramento
  - é programado pela CPU para realizar as transferências



Operações de Entrada e Saída

- Exemplo de Polling no x86

```
WAIT:   in AL, 1              # lê estado do dispositivo (porta 1)
        cmp AL, READY         # compara AL com valor READY
                              # seta Z=(AL == READY)
        jnz WAIT              # se Z≠1 repete
        in AX, 2              # senão lê  dado (porta 2) para AX
```

- Exemplo de Polling no RISC-V

```
.eqv MASK 0x001
.eqv STATUS 0x000
.eqv DATA 0x004
.text
WAIT:   lw  t0, STATUS(s0)   # lê estado do dispositivo s0
        andi t1, t0, MASK    # Isola o bit status por MASK
        beq t1, zero, WAIT   # se não está pronto repete
        lw  s1, DATA(s0)     # senão lê o dado para s1
```

Exceções e Interrupções no RISC-V

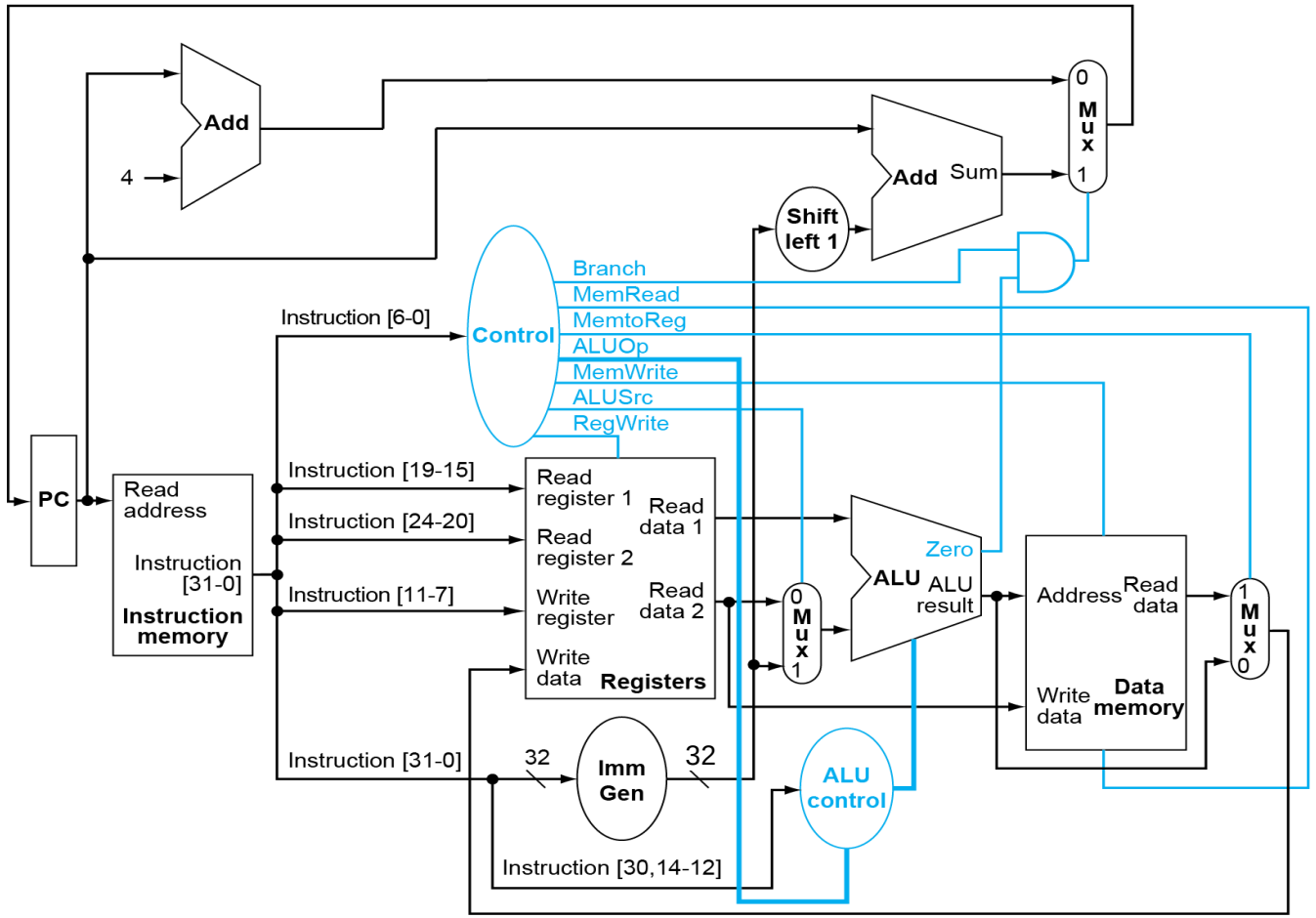
Vamos simplificar para exemplificar a implementação:

Objetivo: Identificar e tratar de

- Exceção de instrução inválida (CAUSE=2)
- Exceção de endereço desalinhado (CAUSE=4 (lw) ou 6 (sw))

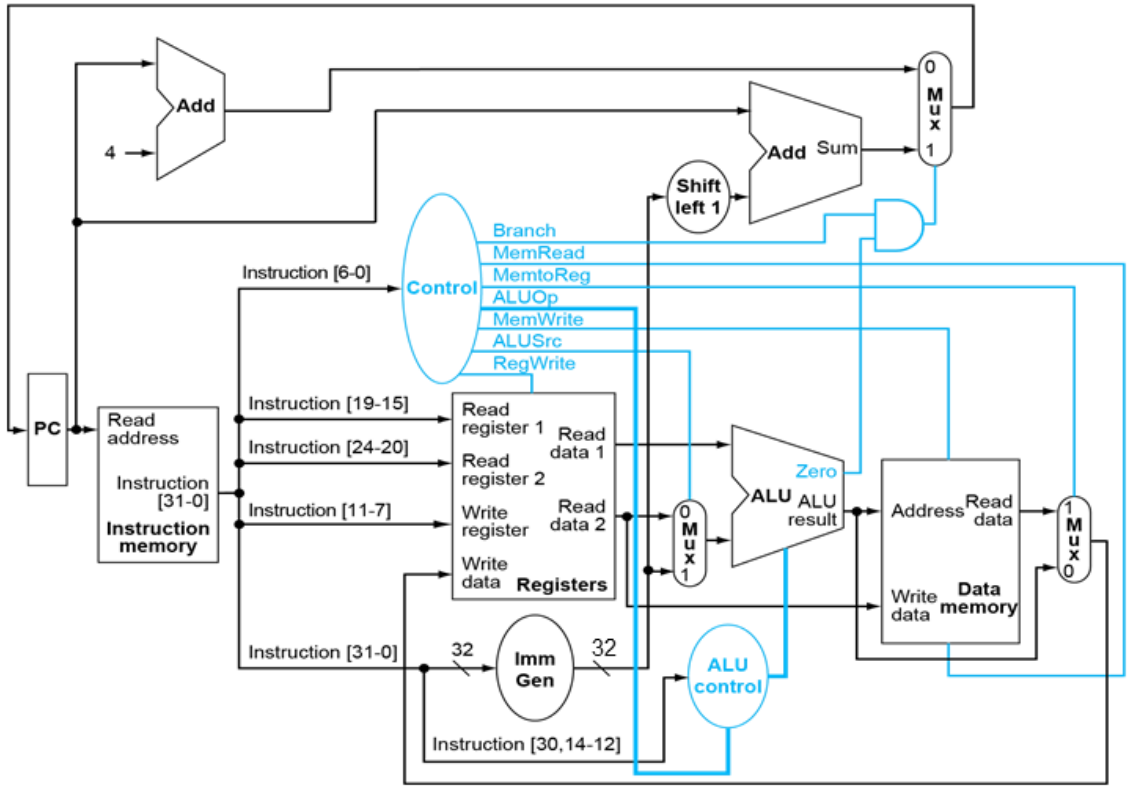
- Acrescentar quatro registradores especiais:
  - **UCAUSE**: indica a causa da exceção/interrupção
  - **UVAL**: contém a instrução inválida ou o endereço desalinhado
  - **UEPC**: indica o endereço da instrução que causou a exceção
  - **UTVEC**: contém o endereço da rotina de tratamento de exceção Ex.: 0x8000000**0**

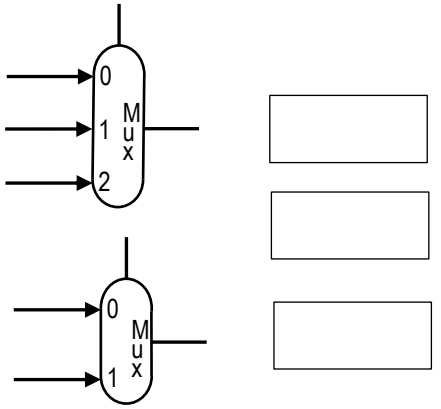
RISC-V Uniciclo



Instrução	ALUSrc	Mem2Reg	RegWrite	MemRead	MemWrite	Branch	ALUOp
Tipo-R	0	0	1	0	0	0	10
lw	1	1	1	1	0	0	00
sw	1	X	0	0	1	0	00
beq	0	X	0	0	0	1	01

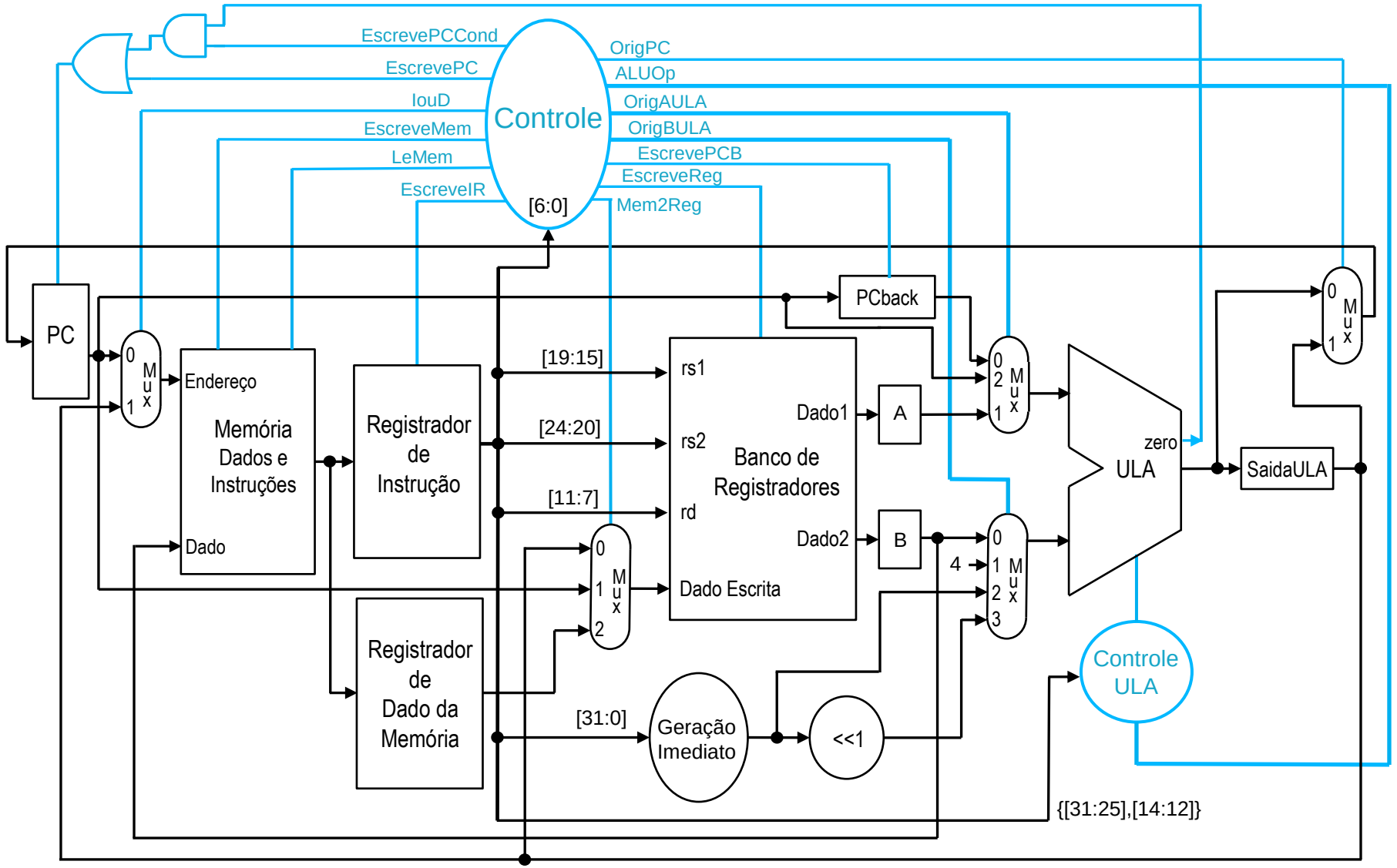
RISC-V Uniciclo



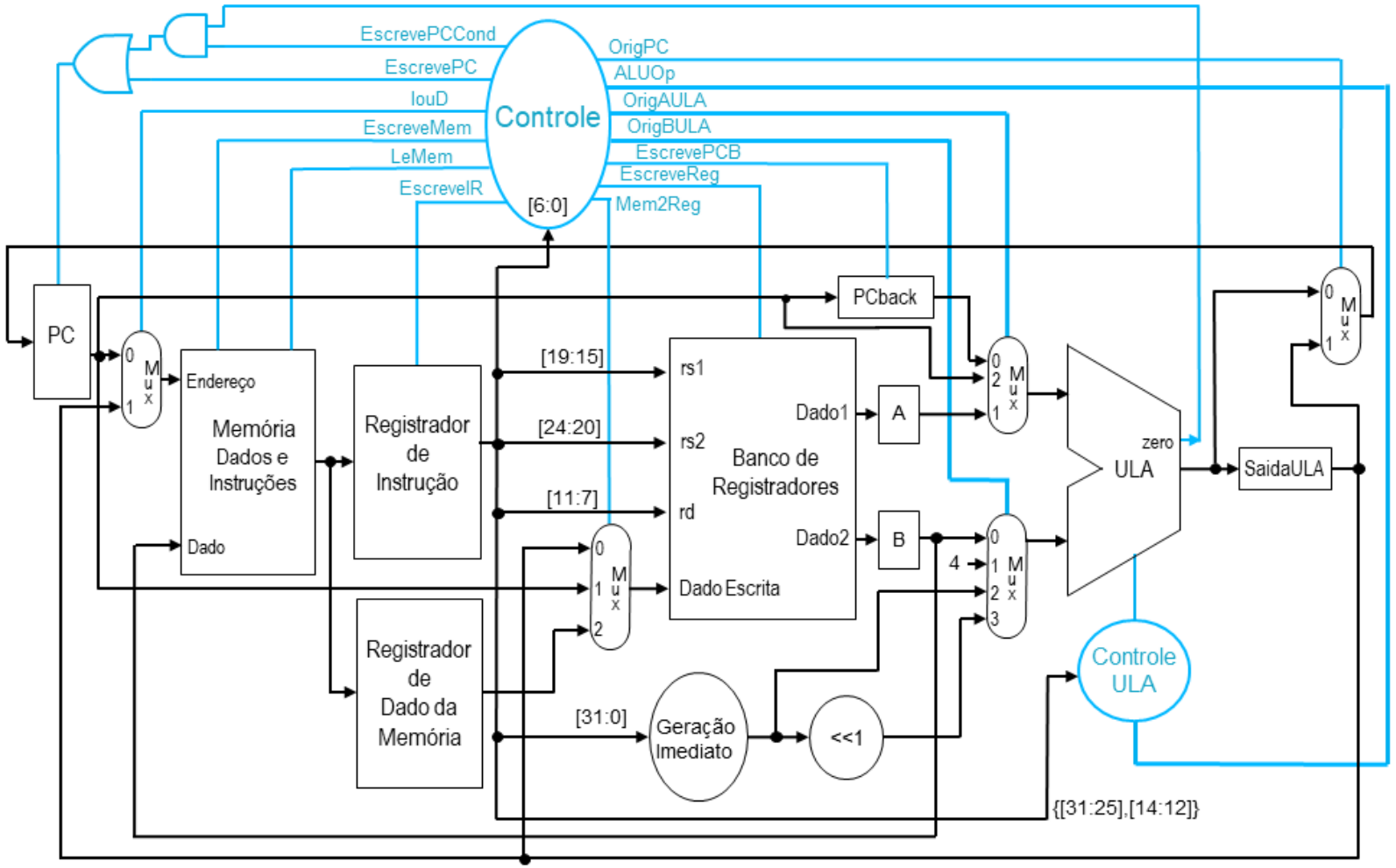


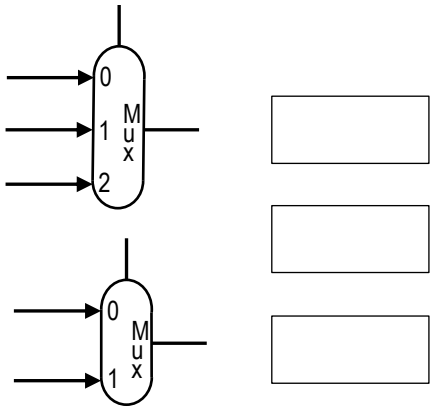
Instrução	ALUSrc	Mem2Reg	RegWrite	MemRead	MemWrite	Branch	ALUOp
Tipo-R	0	0	1	0	0	0	10
lw	1	1	1	1	0	0	00
sw	1	X	0	0	1	0	00
beq	0	X	0	0	0	1	01

RISC-V Multiciclo

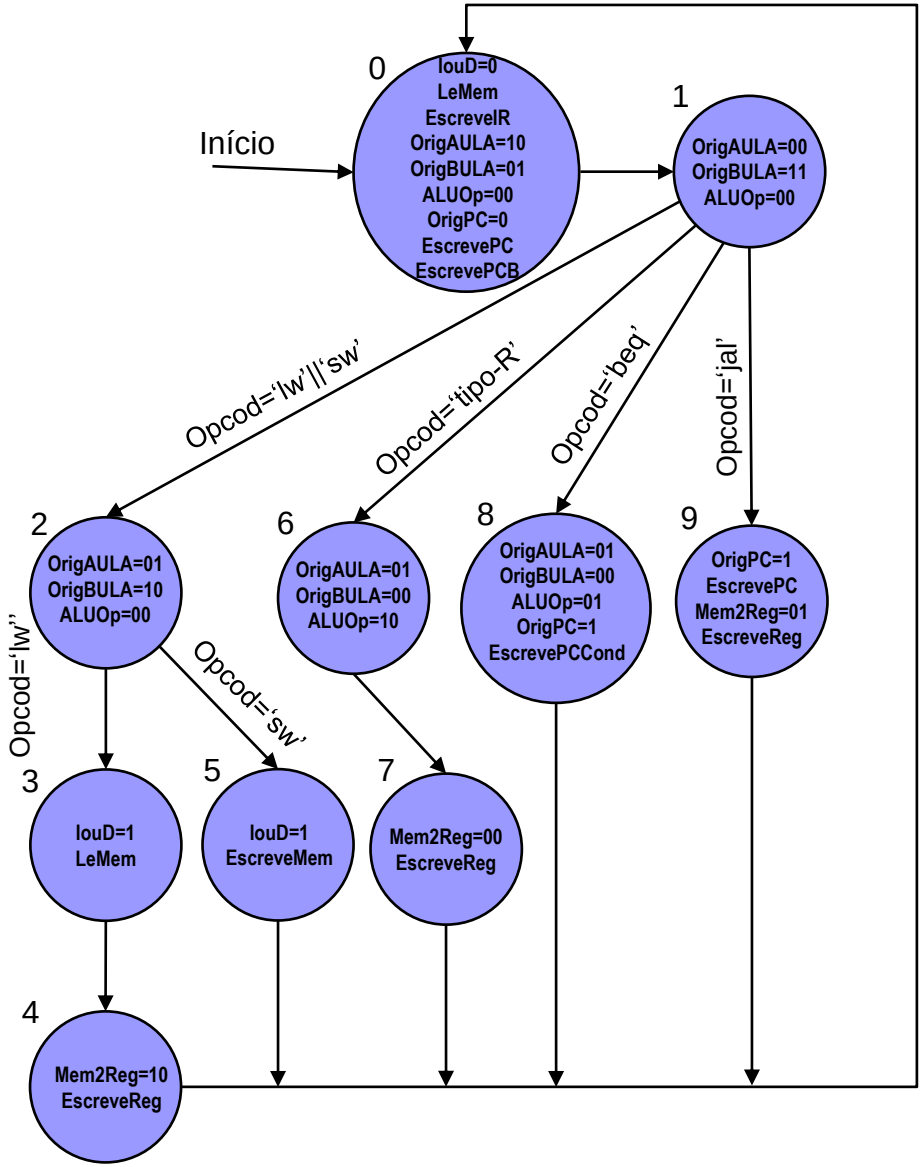


RISC-V Multiciclo

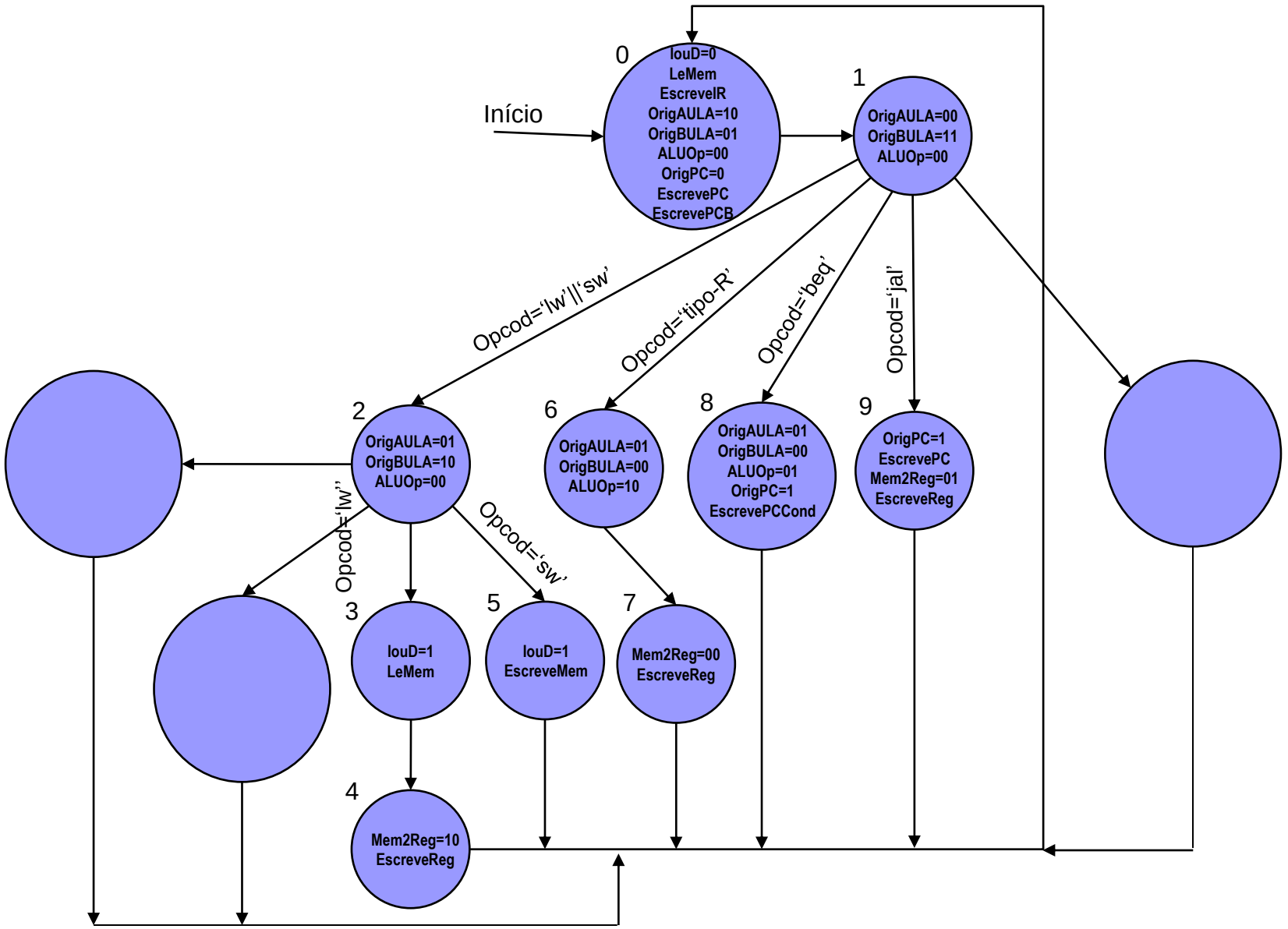




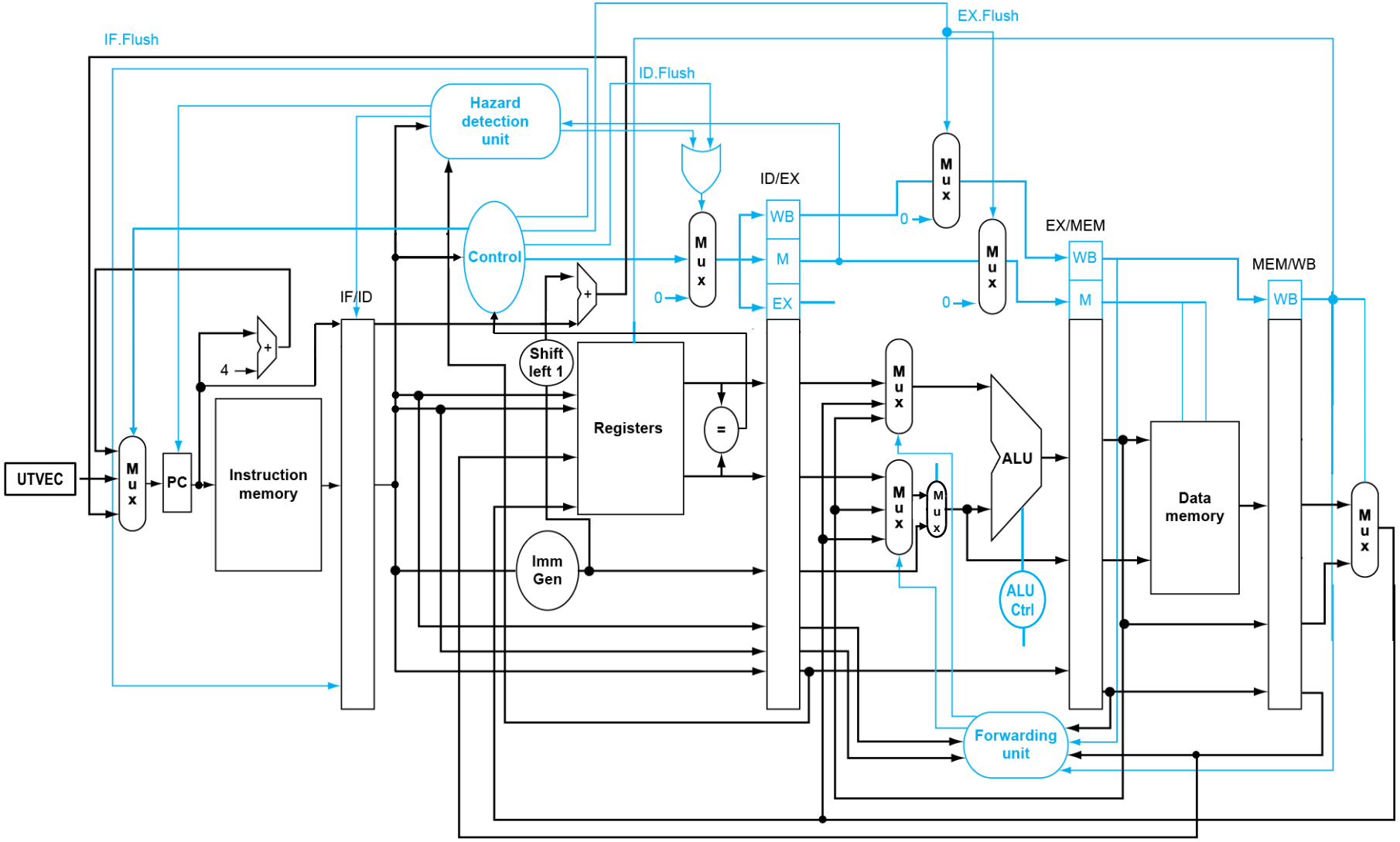
Controle do RISC-V Multiciclo



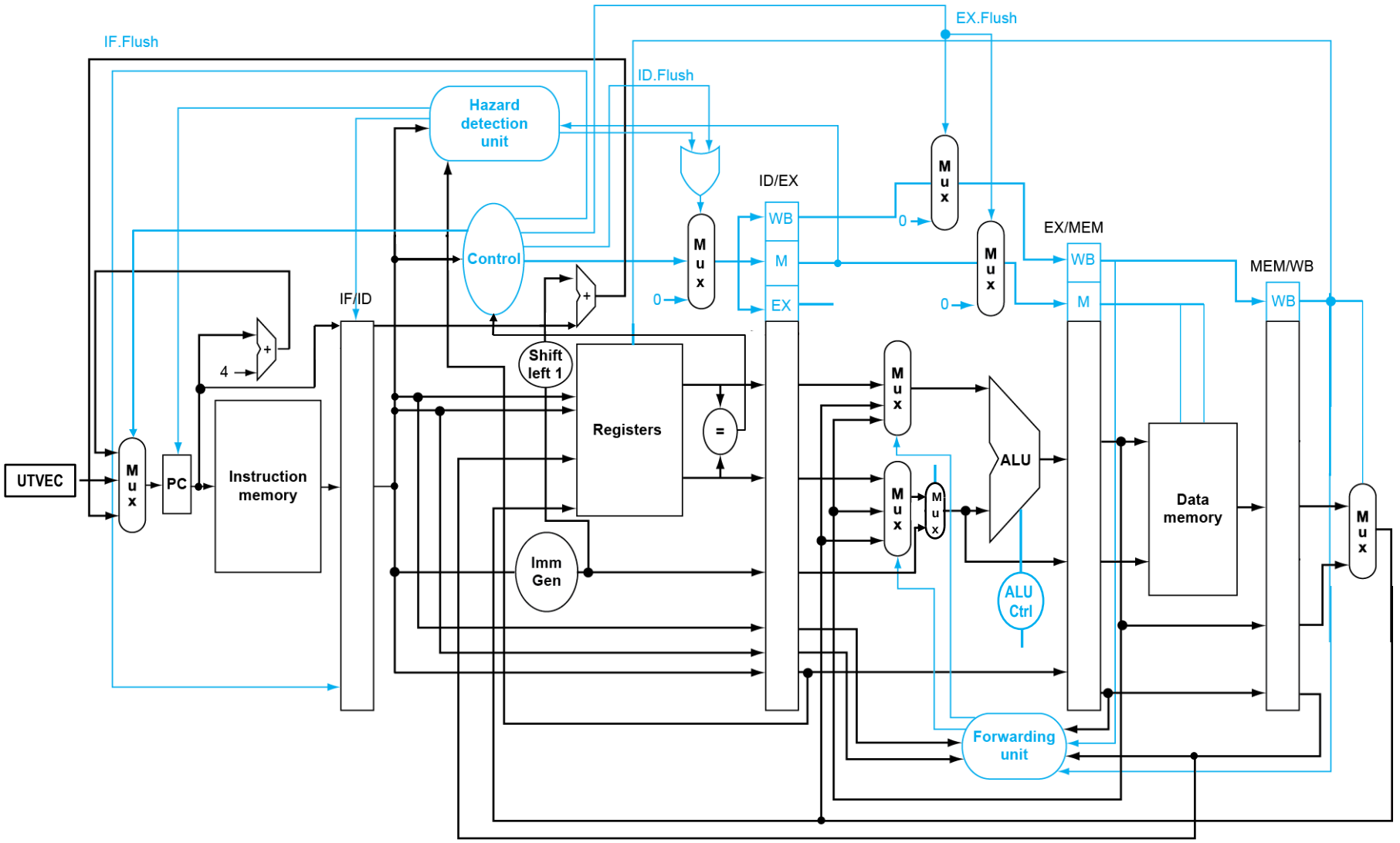
Controle do RISC-V Multiciclo



RISC-V Pipeline



RISC-V Pipeline



Conclusões

Qualquer mudança no fluxo do programa, seja por desvios condicionais e incondicionais, chamadas para o sistema, exceções e interrupções, hazards de controle, são complicadas e devem ser trabalhadas com cuidado no Pipeline.

Pipeline não é uma técnica fácil de se aplicar mas ainda é o estado-da-tecnologia em organização de processadores (99,9% dos processadores comerciais a utilizam).