

## Aula 23

- Excepções e interrupções:
  - Gestão de excepções e interrupções
  - *Datapath* multiciclo com suporte para as excepções provocadas por *overflow* e *opcode* desconhecido
  - A unidade de controlo com suporte para as excepções
- Exemplos da metodologia de instanciação de novas instruções baseadas no datapath multiciclo analisado

Bernardo Cunha, José Luís Azevedo, Arnaldo Oliveira, Tomás Oliveira e Silva

## Excepções e interrupções

- O que são excepções?
- Excepções são eventos que, não sendo *branches* ou *jumps*, alteram o fluxo normal de execução do programa. Existem duas fontes distintas de excepções:
  - Eventos **internos** ao CPU, inesperados e decorrentes da execução das próprias instruções
    - O **overflow** aritmético ou o *fetch* de uma instrução com um **OPCODE desconhecido** para a unidade de controlo são dois exemplos de excepções
  - Eventos **externos** ao CPU que surgem assincronamente com o funcionamento deste. Estes eventos são normalmente designados por **interrupções** (*interrupts*). Tal como as excepções provocam uma alteração do fluxo de execução do programa
    - As fontes de interrupção são normalmente dispositivos de I/O que necessitam de ser atendidos (por exemplo o teclado quando uma tecla é premida)

## Excepções e interrupções

Alguns eventos deste tipo podem ser observados na tabela seguinte.

Tipo de evento	Proveniência	Terminologia MIPS
Pedido de atenção por um dispositivo de I/O	Externo	<i>Interrupt</i>
Invocação do sistema operativo a partir do programa	Interno	Excepção
<b>Overflow aritmético</b>	<b>Interno</b>	<b>Excepção</b>
<b>Tentativa de execução de uma instrução desconhecida</b>	<b>Interno</b>	<b>Excepção</b>
Problema de funcionamento do hardware	Qualquer	Excepção ou <i>interrupt</i>

A detecção das condições de excepção e as acções que daí decorrem podem condicionar a eficiência do CPU ao influenciar directamente o período do sinal de relógio. É, assim, fundamental considerar estas condições desde a fase inicial do projecto da unidade de controlo.

## Excepções – operações básicas efectuadas pelo $\mu P$

- Há duas operações básicas que têm de ser efectuadas pelo processador sempre que ocorre uma excepção:
  1. Copiar o endereço da instrução que gerou a excepção para um registo a isso destinado. No **MIPS** esse registo chama-se **EPC** (Exception Program Counter)
  2. Transferir a execução para a **rotina de tratamento da excepção** (carregando no *Program Counter* um endereço previamente definido)
- A rotina de tratamento da excepção, para além de conhecer o endereço da instrução que gerou a excepção, deverá também poder determinar a causa, por forma a agir em conformidade

## Excepções – Identificação da origem / causa

- Para a identificação da causa da excepção, na respectiva rotina de tratamento, os dois mecanismos mais utilizados são:
  - A existência de um registo especial no qual é armazenado um código que identifica a causa da excepção ou do *interrupt*. Esse registo pode ser acedido pela rotina de tratamento para determinar a causa
    - Este é o mecanismo usado pelo MIPS
    - O registo é designado **Cause Register**
  - A vectorização das excepções
    - Nesta estratégia, cada tipo de excepção (ou *interrupt*) desencadeia, directa ou indirectamente, a passagem da execução do programa para um endereço distinto (e.g. excepção X causa a transferência da execução do programa para o endereço A; excepção Y causa a transferência da execução do programa para o endereço B, ...)

## Excepções – processamento na rotina de atendimento

- A rotina de tratamento da excepção poderá realizar uma de várias tarefas possíveis:
  - Se a excepção corresponde a um evento que compromete a continuação da execução do programa do utilizador, então deverá terminar a execução do mesmo (e.g. acesso a um endereço não alinhado)
  - Providenciar um serviço ao programa do utilizador (syscall)
  - Tomar uma medida de reparação do evento que gerou a excepção (quando seja possível)
  - Retornar ao programa do utilizador, caso a excepção não seja crítica, usando para isso a informação guardada no EPC

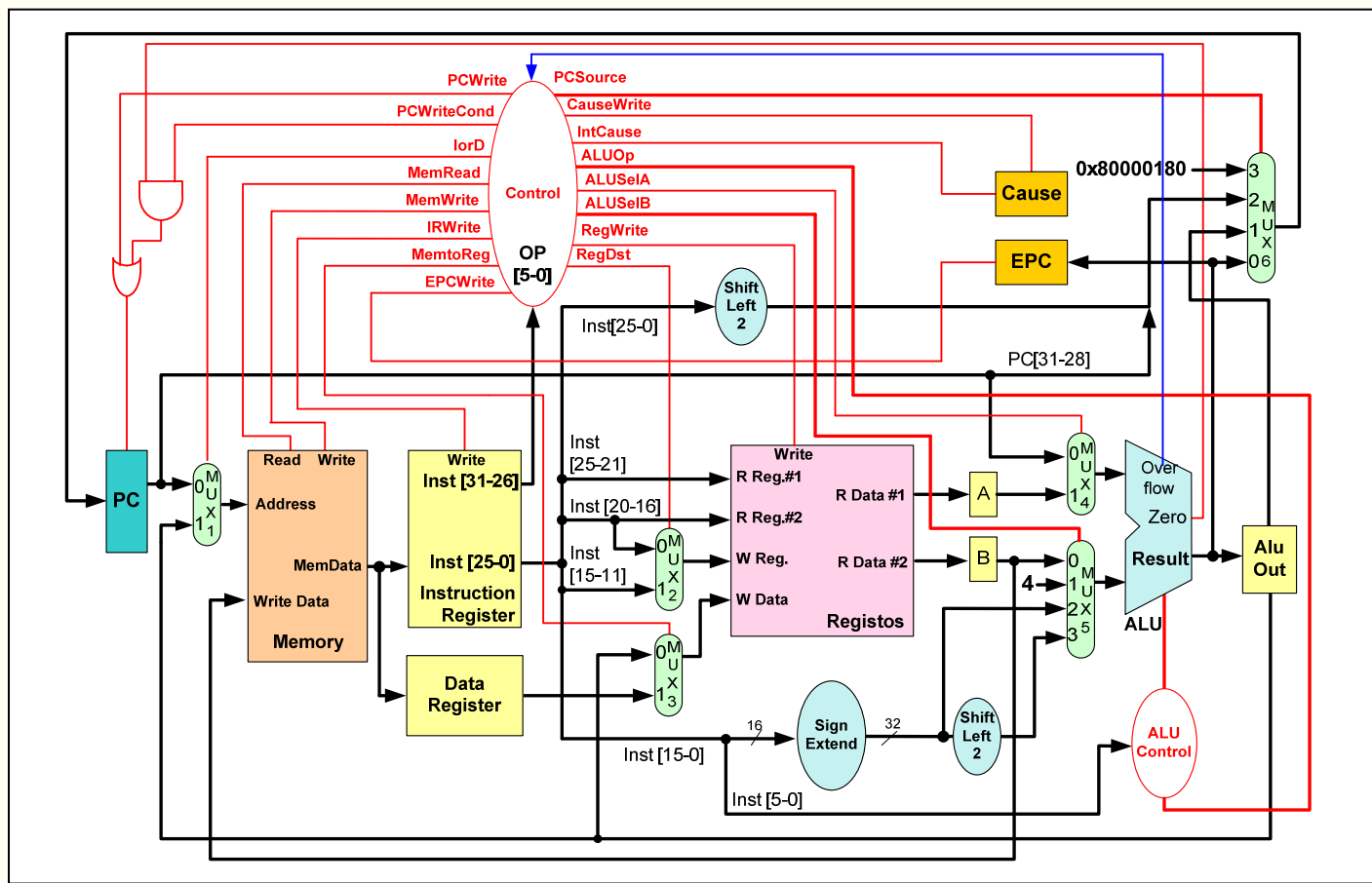
## Exceções – integração no *datapath multicycle* do MIPS

- Na análise que iremos fazer a seguir, vamos apenas considerar dois tipos de exceções:
  - Exceções geradas pela tentativa de execução de uma **instrução inválida** (campo *OPCODE* desconhecido da unidade de controlo)
  - Exceções geradas pela ocorrência de **overflow** na execução de uma operação aritmética
- O mecanismo para atendimento de *interrupts* é semelhante. Há, no entanto, uma diferença fundamental:
  - **Exceções**: a instrução que gerou a exceção não termina, sendo a execução passada de imediato para a rotina de tratamento da exceção
  - **Interrupções**: a passagem para a rotina de tratamento da interrupção só acontece quando for concluída a instrução que está a ser executada no momento em que a interrupção surge

## Exceções – integração no *datapath* multíciclo do MIPS

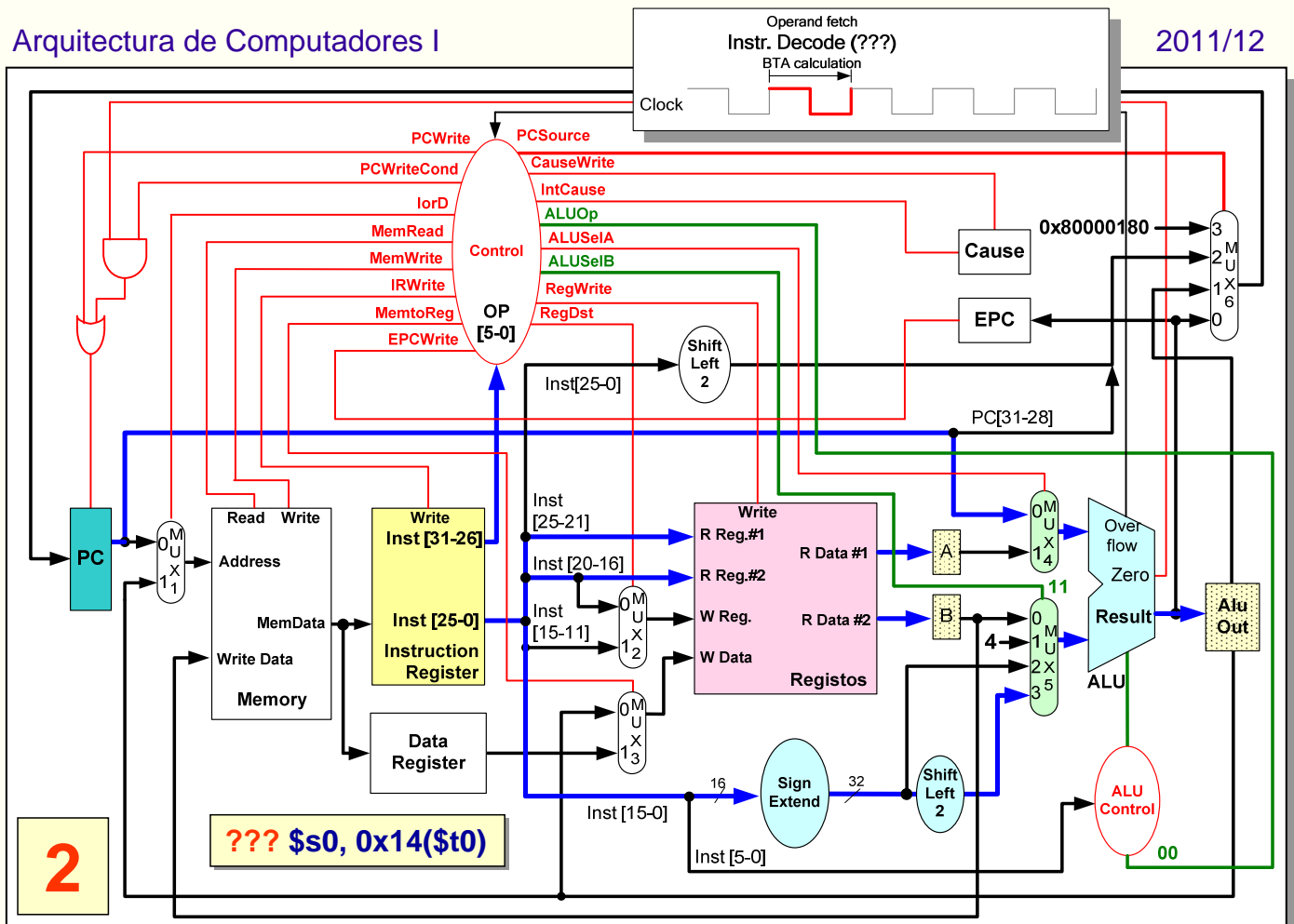
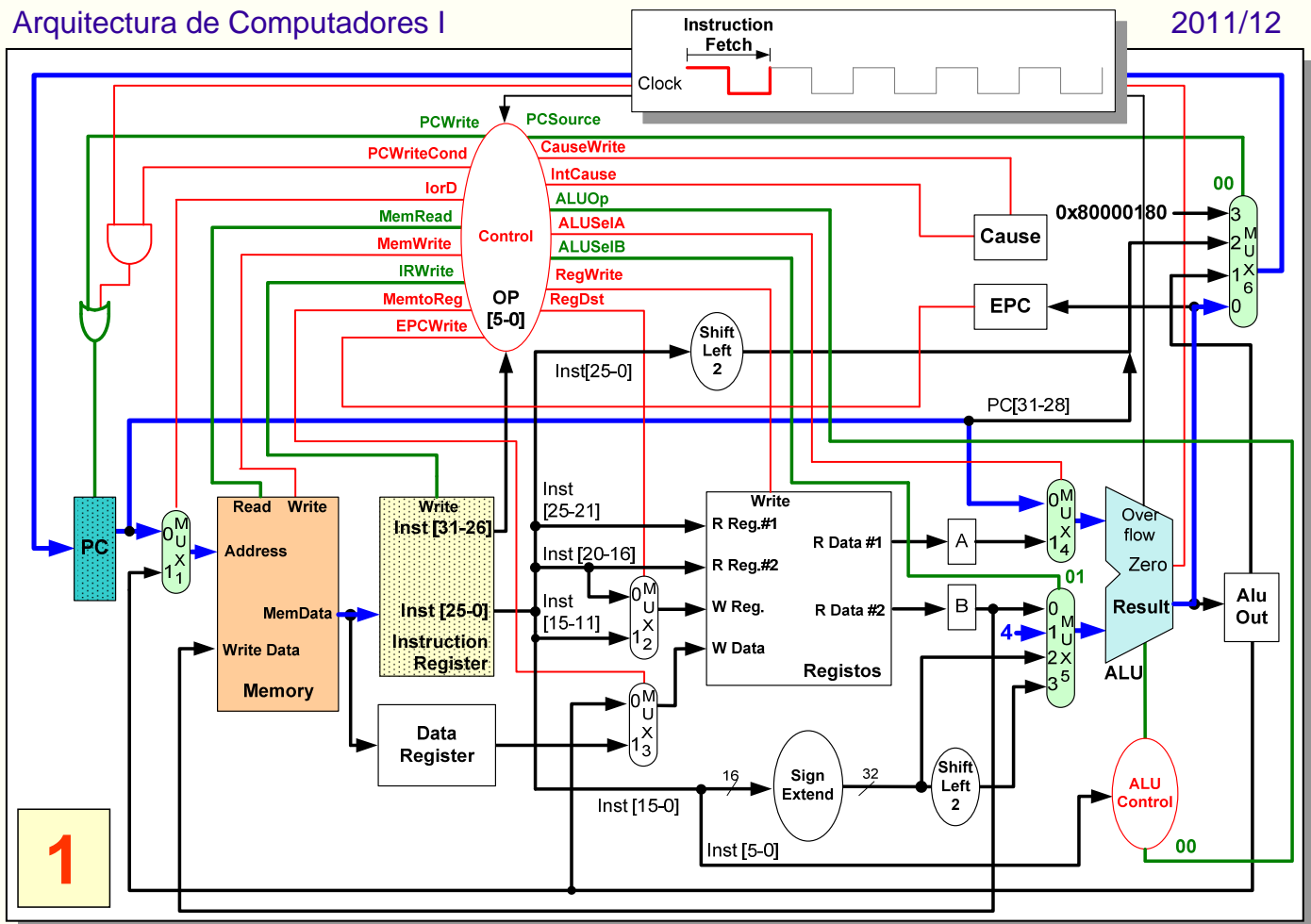
- Para suportar a gestão dos dois tipos de exceção indicados, o nosso *datapath* multi-ciclo precisa de ser ligeiramente alterado. Essas alterações podem ser sintetizadas da seguinte maneira:
  - Criação de dois novos registos – **EPC** e **Cause Register**. No caso deste último, apenas um bit será usado para identificar a causa da exceção:
    - O valor “0” indicará uma instrução desconhecida
    - O valor “1” indicará a ocorrência de um *overflow*
  - Criação de um conjunto suplementar de sinais de controlo que permitam operar sobre aqueles dois registos
  - Criação de um mecanismo que permita escrever no PC o endereço da primeira instrução da rotina de atendimento à exceção (considera-se aqui que esse endereço é **0x80000180**)

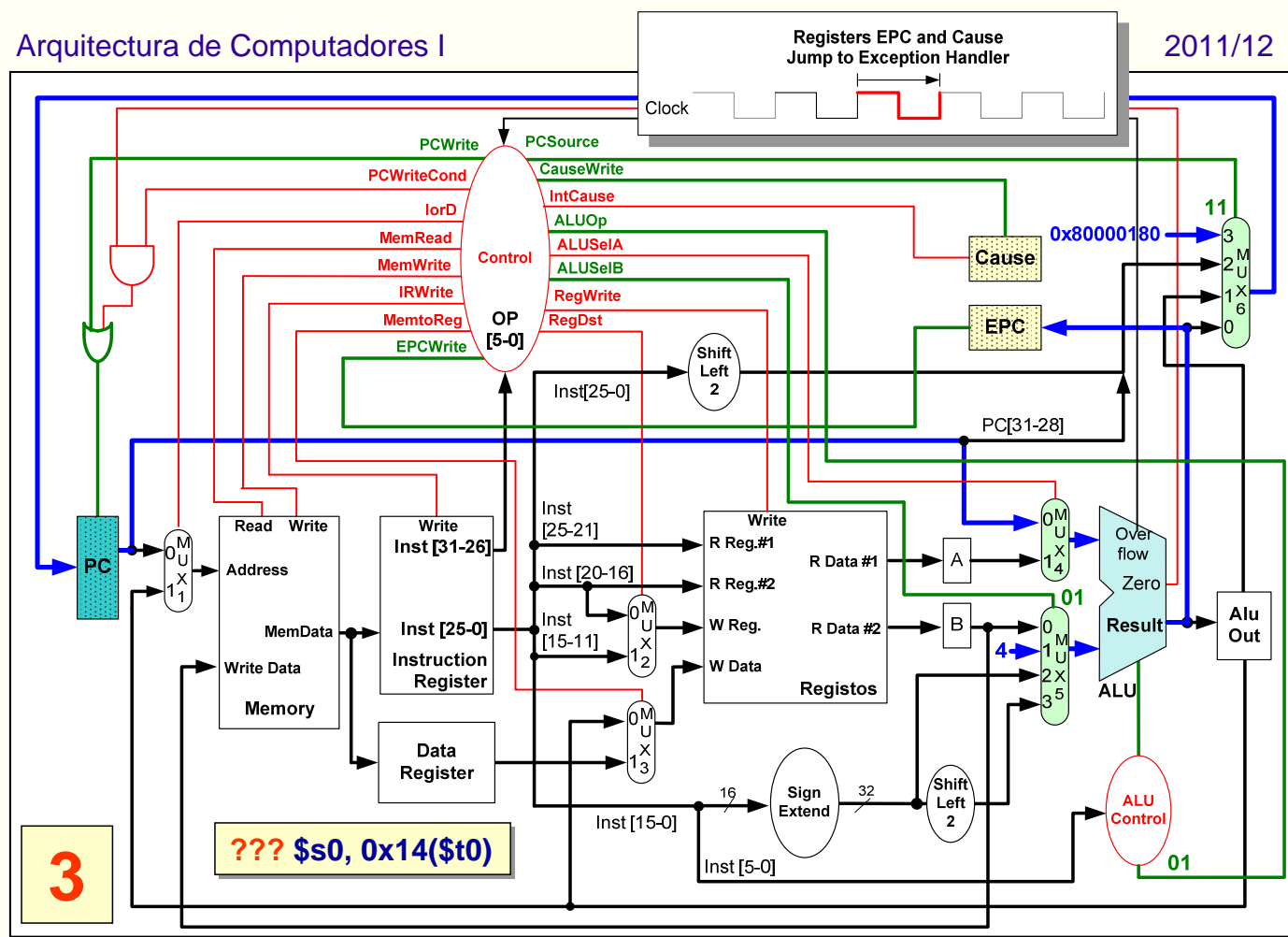
Os registos EPC e Cause podem ser acedidos através do coprocessador 0 do MIPS



## Exemplo 1

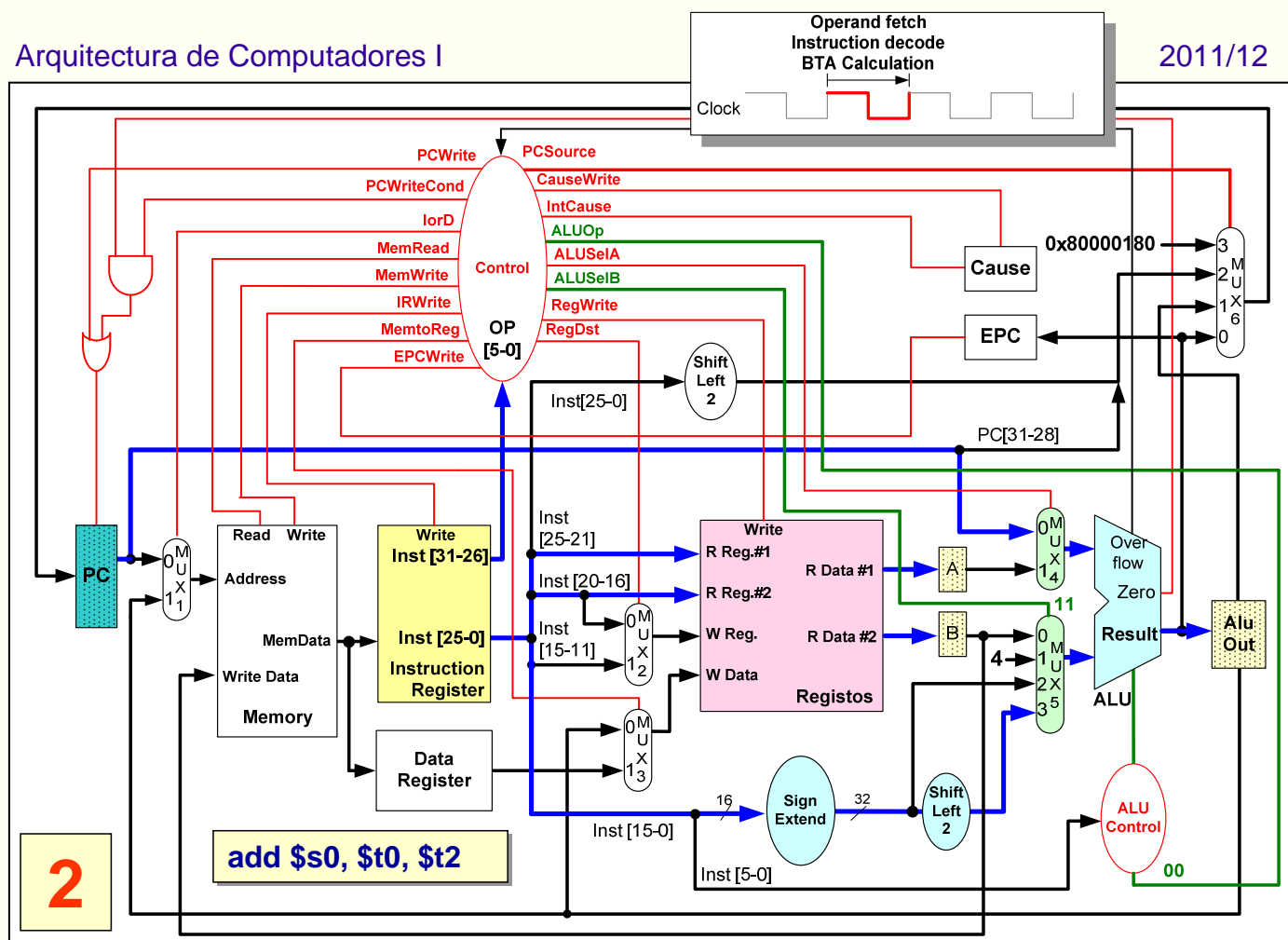
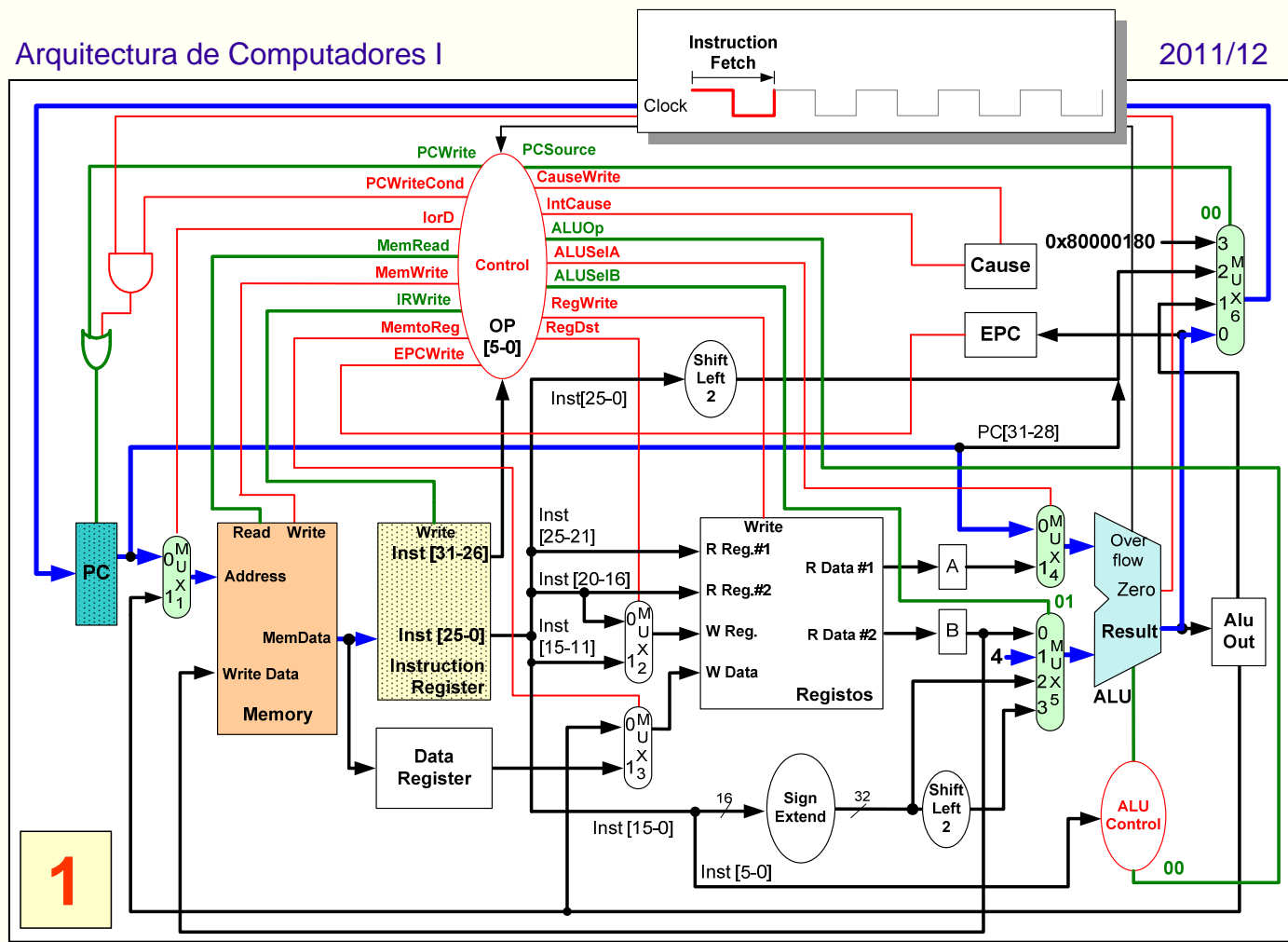
Funcionamento do *datapath* na situação em que o código (*opcode*) da instrução lida é desconhecido





## Exemplo 2

Funcionamento do *datapath* na situação em que uma operação aritmética gera *overflow*







## A unidade de controlo com suporte para as exceções

- A unidade de controlo tem também de ser redesenhada para acomodar a gestão das exceções e a geração dos respectivos sinais de controlo
- Como pudemos observar no *datapath*, a gestão dos dois tipos de excepção, considerados nestes exemplos, carece apenas de um ciclo de relógio suplementar cada um
- Ao nível do diagrama de estados precisaremos portanto de dois estados suplementares

**Exercício:** Faça as alterações ao *datapath* que entender necessárias para que seja gerada uma excepção sempre que o endereço de acesso à memória (nas respectivas instruções) não seja múltiplo de 4. Reflicta essas alterações na unidade de controlo.

