

# **ARM7**

Wesley Nunes Gonçalves

# Instruções Implementadas

- Instruções do tipo processamento de dados.
  - Exemplo: ADD, SUB e CMP.
- Instruções do tipo Multiplicação.
  - Exemplo: MUL e MLA.
- Instruções do tipo *Branch*.
  - Exemplo: B e BL.
- Registrador de Controle CPSR com as *flags*.

# Campo COND

- Utilizado para condicionar a execução de uma instrução.
  - Exemplo:
    - `CMP r1, r2`
    - `B -5 (Cond=0x00)`.

# Exemplo de Programas

## Processamento de Dados:

```
int r1 = 1
int r2 = 3
int r3 = r2*r1
int r4 = r1-r2
```

```
ADD R1, $ZERO, 1
ADD R2, $ZERO, 3
MUL R3, R2, R1
SUB R4, R1, R2
```

## Loop:

```
int fatorial = 1;
int n = 5;
fat: fatorial = fatorial * contador;
contador = contador + 1;
if(contador <= n) GOTO fat;
```

```
ADD r1, $zero, 2
ADD r2, $zero, 1
ADD r3, $zero, 5
MUL r2, r2, r1
ADD r1, r1, 1
CMP r1, r3
B -5
NOP
NOP
```

# Sintaxe

## Branch:

```
int r1 = 3
int r2 = 3
int r3 = 1
if(r1 != r2) {
    r3 = r3 + 2
    r3 = r3 + r2
}
r3 = r3 * r3
```

```
ADD R1, $ZERO, 3
ADD R2, $ZERO, 3
ADD R3, $ZERO, 1
TEQ R1, R2
B 3
NOP
NOP
ADD R3, R3, 2
ADD R3, R3, R2
MUL R3, R3, R3
```

# Pipeline

- ARM7 em três estágios.
  - IF – Instruction Fetch.
  - ID – Instruction Decode.
  - EX – Execute.

# Artigo

- Alocação de registradores com escalonamento de instruções.
- Atualmente alocação é realizada através da coloração dos grafos.
  - Não alcança uma bom uso da máquina.
- Utilização de um escalonador de instrução.

# Artigo

- Em que ordem realizar o escalonamento e a alocação?
  - Antes – Mais registradores são necessários.
  - Depois – Introduz falsas dependências.
- Proposta:
  - Aplicar as duas etapas simultaneamente.



# Algoritmo Proposto

- 1) Geração do grafo de escalonamento.
  - Cada vértice corresponde a uma instrução.
  - Aresta  $u$  e  $v$  se:
    - Há uma dependência de dados de  $v$  para  $u$ .
    - Há uma dependência de controle de  $u$  para  $v$ .
    - Há uma quantidade de unidades funcional que força a precedência de  $u$  sobre  $v$ .
- 2) Calcular  $EP(v)$  para cada vértice do grafo de escalonamento.  $EP(v)$  representa o menor tempo em que uma instrução pode ser escalonada.

# Algoritmo Proposto

- 3) Construção de um grafo de interferência
  - Inclui o grafo de falsas dependências e o grafo de interferência.
    - Cada vértice corresponde a definição de variável.
    - Aresta  $u$  e  $v$  se uma definição está viva até a definição da outra.

# Sintaxe

- 4) Coloração através do número de registradores e os grafos calculados em 1 e 3.

# Algoritmo e ARM7

- O algoritmo pode ser aplicado ao processador ARM7.
  - Apresenta uma boa solução para escalonamento das instruções com alocação de registradores.
  - Auxilia a execução em paralelo das instruções.
    - Por exemplo a instrução de desvio.

# Conclusão e Trabalhos Futuros

- Implementação de uma versão simplificada do processador ARM7.
- Registrador de Status.
- Alocação de registradores com escalonamento de instrução.
- Implementação do comportamento das instruções de Load e Store.
- Implementar as instruções

# Obrigado

- [wnunesgoncalves@gmail.com](mailto:wnunesgoncalves@gmail.com)