PCS-3225 Sistemas Digitais II — 24 de abril de 2024 - v4

Instruções de Desvio

Aula 12 - Exercício

Bruno Albertini, Edson Gomi e Ricardo Lera

O objetivo deste exercício é entender como *loops* são implementados em código Assembly do RISC-V e praticar o uso das instruções de desvio condicional, especificamente bqe e bne.

Traduza o seguinte programa em C para Assembly do RISC-V:

```
i = 0;
b = 7;
while (a[i] == b)
  s = a[i] + i;
  i = i + 1;
```

O vetor a[ ] está armazenado a partir do endereço 0x50 da memória de dados do RISC-V e contém uma quantidade não divulgada de números 7, seguido de um número diferente de 7. A variável s está no endereço 0x40.

O RISC-V colocado no Juiz está preparado para executar as seguintes instruções:

- add rd, rs1, rs2
- sub rd, rs1, rs2
- addi rd, rs1, im // im = imediato
- lw rd, im(rs1)
- sw rs1, im(rs2)
- beq rs1, rs2, im
- bne rs1, rs2, im

A Figura 1 apresenta os formatos das instruções a serem usadas neste exercício.

Name	Field						Comments
(Field size)	7 bits	5 bits	5 bits	3 bits	5 bits	7 bits	
R-type	funct7	rs2	rs1	funct3	rd	opcode	Arithmetic instruction format
I-type	immediate[11:0]		rs1	funct3	rd	opcode	Loads & immediate arithmetic
S-type	immed[11:5]	rs2	rs1	funct3	immed[4:0]	opcode	Stores
SB-type	immed[12,10:5]	rs2	rs1	funct3	immed[4:1,11]	opcode	Conditional branch format

Figura 1: Codificação das Instruções do RISC-V

A Tabela 2 apresenta os formatos das instruções disponíveis neste exercício.

O RISC-V colocado no Juiz para este exercício tem as seguintes características diferentes de um RISC-V padrão:

Instrução	funct7	funct3	opcode
add	0000000	000	0110011
sub	0100000	000	0110011
addi	-	000	0100011
lw	-	010	0000011
sw	-	010	0100011
beq	-	000	1100011
bne	-	001	1100111

Figura 2: Formatos das Instruções deste exercício

1. Os dados são de 64 bits e cada palavra da memória RAM tem 64 bits de tamanho. O endereçamento é feito palavra a palavra e não byte a byte;

2. As palavras da memória ROM de instruções têm 32 bits de tamanho e também são endereçadas palavra a palavra;

3. As instruções de desvio condicional beq e bne usam o formato S ao invés do formato SB do RISC-V padrão. Considere esta informação ao calcular o imediato do endereçamento relativo.

Para facilitar a codificação do programa Assembly, sugerimos o uso de uma ferramenta decodificadora de instruções do RISC-V. Aviso importante: como o processador colocado no Juiz para este exercício não é RISC-V padrão, a codificação dada por esta ferramenta não funciona para as instruções de desvio condicional.

Faça o download do arquivo rom.v que está disponível no e-Disciplinas. Preencha a ROM com as instruções do seu programa Assembly. Faça o upload do arquivo rom.v no Juiz. O Juiz verificará se o valor da soma contida na variável s na RAM está correto face à quantidade de números 7 armazenados na memória RAM.

Cada integrante do time terá 10 tentativas no Juiz.

A memória ROM implementada no Juiz tem tamanho de 32 palavras de 32 bits. O seu programa não pode ter mais que 32 instruções. Aqui surge uma pergunta: qual deve ser a instrução seguinte à última instrução do seu programa, isto é, o que o processador deve fazer após executar a última instrução do seu programa? O fato é que o processador continuará a executar instruções enquanto estiver ligado. Uma solução é fazer com que o processador pare de executar instruções. No RISC-V podemos usar a instrução wfi (wait for interrupt). Assim, a recomendação é que preencha a ROM com a instrução wfi nas palavras após a última instrução do seu programa.

Há outra pergunta: o que acontece com o processador se ele encontra uma palavra que não corresponde a uma instrução válida?