

Projeto ISA e Microarquitetura

Gabriel Bardini Bessa

25 de novembro de 2024

1 Definição da ISA

A ISA (Instruction Set Architecture) definida neste trabalho é baseada em um conjunto de instruções de 24 bits, organizadas em três formatos principais: R, I e J. O objetivo é criar uma arquitetura capaz de realizar operações aritméticas, lógicas, de memória e de controle.

1.1 Formatos de Instrução

Os formatos de instrução foram definidos como segue:

Formato	Tamanho	Descrição
R	24 bits	Operações aritméticas e lógicas entre registradores.
I	24 bits	Operações com valores imediatos ou acesso à memória.
J	24 bits	Instruções de salto e controle de fluxo.

1.2 Conjunto de Instruções

As instruções suportadas pela ISA incluem:

Formato Descrição

Nome	Formato	Descrição
ADD	R	Soma dois registradores.
SUB	R	Subtrai dois registradores.
MUL	R	Multiplica dois registradores.
DIV	R	Divide dois registradores.
ADDI	I	Soma um valor imediato ao registrador.
BNE	I	Salta se dois registradores forem diferentes.
BEQ	I	Salto se dois registradores forem iguais.
JUMP	J	Salto incondicional
AND	J	Realiza uma operação AND bit a bit entre dois valores
OR	J	Realiza uma operação OR bit a bit entre dois valores
XOR	J	Realiza uma operação XOR bit a bit entre dois valores
ANDI	J	Realiza uma operação AND bit a bit entre um valor e um imediato
LW	I	Carrega uma palavra da memória para o registrador
SW	I	Salva uma palavra da memória para o registrador

2 Implementação da Microarquitetura

A microarquitetura foi implementada no Logisim Evolution, com os seguintes componentes principais:

Componente	Descrição
PC	Contador do programa de 24 bits
Banco de Registradores	8 registradores de 24 bits.
ULA	Unidade Lógica e Aritmética para operações R e I.
Memória de Instruções	Armazena instruções de 24 bits.
Memória de Dados	Armazena palavras de 24 bits alinhadas.

O funcionamento da microarquitetura segue os passos clássicos de busca, decodificação e execução de instruções.

3 Código Assembly

O seguinte código assembly foi desenvolvido para testar as instruções da ISA:

```
# Soma dois valores e armazena na memória
ADDI$4, $0,10      #Carrega o valor 10 no registrador $4
ADDI$5, $0,20      #Carrega o valor 20 no registrador $5
ADD$6, $4,$5        #Soma $4 e $5, armazena o resultado em $6
SW$6, 0($8)         #Salva o resultado na memória
```

A tradução para linguagem de máquina segue o formato definido:

Assembly	Binário
ADDI \$4, \$0, 10	000001 00000 00100 000000001010
ADD \$6, \$4, \$5	000000 00100 00101 00110 000 000
SW \$6, 0(\$8)	000010 00110 01101 000000000000