

Especificação Processador RISC16

O processador RISC16 foi especificado dentro do projeto do Milenium, com o objetivo de ser integrado em um sensor inteligente *wireless* para controle de irrigação. O texto a seguir descreve a arquitetura do seu conjunto de instruções.

1. Instruções e Registradores

Um conjunto de 16 instruções de tamanho fixo foi desenvolvido para programar o processador em questão. Essas instruções, esquematizadas na figura 1, podem ser agrupadas em três formatos distintos:

- **R** – especifica três registradores;
- **I** – especifica dois registradores e uma constante de quatro bits;
- **J** – especifica um registrador e uma constante de oito bits.

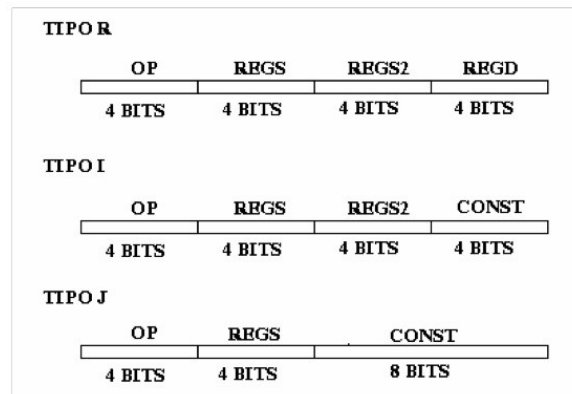


Figura 1. Formato das Instruções do RISC16.

Conforme pode ser visto na figura 1 as instruções podem ser divididas em até quatro campos os quais são definidos abaixo:

- **Operação (OP):** campo de quatro bits que especifica o código da instrução;
- **Registrador fonte (REGS/REGS2):** campo de quatro bits que especifica o número do registrador do banco a ser usado como operando da ULA;
- **Registrador destino (REGD):** campo de quatro bits que especifica o registrador que armazena o resultado da operação da ULA.
- **Constante (CONST):** Identifica um valor constante de quatro ou oito bits.

O conjunto proposto pode ser ainda dividido em quatro categorias: Aritmética, lógica, transferência e desvio. Na tabela 1 estão listadas as instruções que o compõem, bem como seu código, formato e descrição.

Algumas instruções são facilmente compreendidas na tabela acima como: Add, Sub, And, Or, Not, Xor, Lw, Sw e Slt. Para as demais, vale a descrição abaixo:

- **Addi** – Instrução de soma com constante de oito bits. Esta instrução é utilizada para armazenar uma constante de até oito bits nos bits menos significativos de um registrador. Pode também ser utilizada em conjunto com a instrução Lui (*load upper immediate*) para armazenar uma constante de até dezesseis bits.

Tabela 1. Conjunto de instruções do RISC16.

Tabela 1: Conjunto de instruções do RISC-V						
COD.	CAT.		INST.	Exemplo	Descrição	Tipo
0010	Aritmética		Add	Add \$s1, \$s2, \$s3	Adiciona \$s1 a \$s2 e armazena em \$s3	R
0011			Sub	Sub \$s1, \$s2, \$s3	Subtrai \$s1 a \$s2 e armazena em \$s3	R
1000			Addi	Addi \$s1, 100	Adiciona \$s1 a constante e armazena em \$s1	J
1001			Shift	Sft \$s1, 8	Desloca \$s1 da constante e armazena em \$s1. Se o valor da constante for negativo, desloca à esquerda.	J
0100	Lógica		And	And \$s1, \$s2, \$s3	AND booleano bit a bit entre \$s1 e \$s2 e armazena em \$s3	R
0101						
1010			Or	Or \$s1, \$s2, \$s3	Or booleano bit a bit entre \$s1 e \$s2 e armazena em \$s3	R
0110			Not	Not \$s1	NOT booleano bit a bit de \$s1 e armazena em \$s1	J
0111			Xor	Xor \$s1, \$s2, \$s3	XOR booleano bit a bit de \$s1 e \$s2 e armazena em \$s3	R
			Slt	Slt \$s1, \$s2, \$s3	Torna \$s3 = 1 se \$s1 < \$s2 senão \$s1 = 0	R
0000	Transferência		Lw	Lw \$s1,\$s2,\$s3	Carrega palavra armazenada no endereço \$s1 deslocado de \$s2 e salva em \$s3	R
0001						
1011			Sw	Sw \$s1,\$s2,\$s3	Carrega palavra armazenada em \$s3 e salva no endereço \$s1 deslocado de \$s2	R
		Lui	Lui \$s1, 100	Carrega a constante nos oito bits mais significativos de \$s1	J	
1100	Desvio	Cond.	Beq	Beq \$s1,\$s2, 5	Se \$s1 = \$s2 desvia programa para \$pc + CONST	I
1101			Blt	Blt \$s1, \$s2, 5	Se \$s1 < \$s2 desvia programa para \$pc + CONST	I
1110	Desvio		J	J \$s1, 100	Desvia para o endereço \$s1 deslocado da constante	J
1111			Jal	Jal \$s1, 100	Desvia para o endereço \$s1 deslocado da constante salvando origem em \$ra	J

- **Shift** – Instrução de deslocamento. Usada para promover um deslocamento no conteúdo do registrador indicado em “REGS” (Tipo J). O valor colocado no campo “CONST” armazena uma constante que possui significação composta: o bit mais significativo indica o sentido do deslocamento (1- à esquerda e 0- à direita), e os quatro menores contém o número de saltos.
- **Lui (load upper immediate)** – Instrução de carregamento de constante nos oito bits mais significativos de um registrador. Por se tratar de uma instrução do tipo J, indica um registrador e uma constante de oito bits. A instrução carrega esta constante nos bits mais significativos do registrador indicado preenchendo com oito zeros os bits menos significativos. É utilizada em conjunto com a instrução addi para carregar uma constante com mais de oito bits num registrador.
- **Beq e Bltn** – Instruções de desvio condicional. Estas instruções comparam o conteúdo de dois

registradores e desviam o programa para o endereço seguinte (\$pc) acrescido do valor da constante indicada, se o teste resultar verdadeiro.

- **J e Jal** – Instruções de desvio incondicional. Estas instruções são utilizadas para promover desvios no programa independente de condições. Por se tratarem de instruções do tipo J, possuem três campos. A instrução possui uma constante que indica o deslocamento a partir do endereço base, e REGS funciona como um ponteiro para esse endereço. O que diferencia as duas instruções é que a instrução *Jal* salva o endereço da instrução seguinte no registrador \$ra e é normalmente usada na execução de sub-rotinas. O valor armazenado em \$ra é usado para retornar para o programa principal. As instruções de salto para registrador foram suprimidas explicitamente, por questões de simplificação, mas seus efeitos podem ser obtidos informando o registrador no campo “REGS” e preenchendo com zeros o campo “CONST”.

Para este projeto, quatro tipos de endereçamento foram previstos e estão listados a seguir:

- **Endereçamento a registrador:** O operando está em um registrador;
- **Endereçamento relativo ao \$pc:** onde o operando está armazenado no endereço formado pela soma do conteúdo do \$pc com uma constante obtida na própria instrução;
- **Endereçamento via registrador -base ou via deslocamento:** onde o operando está no endereço de memória obtido a partir da soma do conteúdo do registrador-base com uma constante armazenada na instrução;
- **Endereço Imediato:** onde o operando é uma constante, dentro da própria instrução.

O processador possui 16 registradores cujas funções estão descritas na Tabela 2.

Tabela 2. Registradores RISC16

CÓDIGO	SÍMBOLO	FUNÇÃO	OBS
0000	\$zero	Constante 0	Constante 0 de 16bits
0001	\$t0	Temporários	Regist. Auxiliares
0010	\$t1		
0011	\$t2		
0100	\$a0	Argumento	Argumentos para operações aritméticas e procedimentos
0101	\$a1		
0110	\$a2		
0111	\$s0	Salvos	Armazena valores durante chamadas de procedimento
1000	\$s1		
1001	\$s2		
1010	\$s3		
1011	\$int	Interrupção	Contém o status das interrupções e o endereço da instrução com erro
1100	\$gp	Apontador Global	Aponta para as variáveis globais no interior da pilha
1101	\$sp	Apontador Pilha	Aponta para o topo da pilha
1110	\$pc	Contador de Prog.	Aponta para a próxima instrução
1111	\$ra	End. de Retorno	Aponta para o endereço de retorno de uma sub-rotina