

Guará Focinho Longo (VLIW)

VLIW Architecture				
Opcode	Tipo	Menemonico	Nome	Operação
Controle				
0000	R	brzr	Branch On Zero Register	if (R[ra] == 0) PC = R[rb]
0001	I	brzi	Branch On Zero Immediate	if (R[0] == 0) PC = PC + Imm.
0010	R	jr	Jump Register	PC = R[rb]
0011	I	ji	Jump Immediate	PC = PC + Imm.
Dados				
0100	R	ld	Load	R[ra] = M[R[rb]]
0101	R	st	Store	M[R[rb]] = R[ra]
0110	I	movh	Move High	R[0] = {Imm., R[0](3:0)}
0111	I	movl	Move Low	R[0] = {R[0](7:4), Imm.}
Aritmética				
1000	R	add	Add	R[ra] = R[ra] + R[rb]
1001	R	sub	Sub	R[ra] = R[ra] - R[rb]
Lógica				
1010	R	and	And	R[ra] = R[ra] & R[rb]
1011	R	or	Or	R[ra] = R[ra] R[rb]
1100	R	not	Not	R[ra] = ! R[rb]
1101	R	slr	Shift Left Register	R[ra] = R[ra] << R[rb]
1110	R	srr	Shift Right Register	R[ra] = R[ra] >> R[rb]
NOP = Free Slot				
1111	R	nop	No Operation	
			VLIW	4 Lanes Fixos
			4 Lanes = 8*4 bits	1° LD / ST / MOV
			4 Regs GP (General Purpose)	2° BR / JUMP
				3° ULA
				4° ULA

- A memória pode ser endereçada em 4 bytes para facilitar a obtenção dos dados.
- Podemos ter 1 controle por FU a fim de reduzir a complexidade do controle.
- Consideramos que não haverão instruções dependentes na mesma palavra e que o compilador não permitirá duas ou mais operações de escrita ao mesmo registrador na mesma instrução.