

RESUMO DAS INSTRUÇÕES DE DESVIO INCONDICIONAL					
INSTRUÇÃO	CÓDIGO	NOME	FORMATO	OPERAÇÃO	OBSERVAÇÕES
B		Unconditional Branch	B offset	<i>branch</i>	A instrução real é interpretada pelo hardware como BEQ r0, r0, offset.
BAL		Branch and Link	BALL offset	<i>procedure_call</i>	Faz uma chamada de procedimento incondicional relativa ao PC
BALC		Branch and Link, Compact	BALC offset	<i>procedure_call (no delay slot)</i>	Faz uma chamada de procedimento incondicional relativa ao PC. A instrução após o desvio NÃO é executada quando o desvio é tomado.
BC		Branch, Compact	BC offset	$PC = PC + 4 + \text{sign_extend}(\text{offset} \ll 2)$	Desvios compactos não têm slot de atraso: a instrução após o desvio NÃO é executada quando o desvio é tomado
BC1EQZ		Branch if Coprocessor 1 (FPU) Register Bit 0 is Equal to Zero	BC1EQZ ft, offset	<i>if FT & 1 = 0 then branch</i>	Desvia se o Registrador do Coprocessador 1 (FPU) Bit 0 for Igual OU Diferente de Zero. A condição é verdadeira se e somente se o bit 0 do registrador FT for zero. Se a condição for falsa, o desvio não é obtido e a execução continua com a próxima instrução.
BC1NEZ		Branch if Coprocessor 1 (FPR) Register Bit 0 is Not Equal to Zero	BC1NEZ ft, offset	<i>If FT & 1 != 0 then branch</i>	Desvia se o Registrador do Coprocessador 1 (FPU) Bit 0 for Igual OU Diferente de Zero. A condição é verdadeira se e somente se o bit 0 do registrador FT for diferente de zero. Se a condição for falsa, o desvio não é obtido e a execução continua com a próxima instrução.
BC1F		Branch on FP False	BC1F cc, offset	<i>if FPConditionCode(cc) = 0 then branch</i>	Testa um código de condição FP e faz um desvio condicional relativo ao PC. Se o bit do código de condição FP cc for falso (0), o programa desvia para o endereço de destino efetivo depois que a instrução no slot de retardo é executada.
BC1FL		Branch on FP False Likely	BC1FL cc, offset	<i>if FPConditionCode(cc) = 0 then branch_likely</i>	Testa um código de condição FP e faz um desvio condicional relativo ao PC. Executa a instrução no slot de atraso apenas se o desvio for tomado. Se o bit cc do código de condição FP for falso (0), o programa desvia para o endereço de destino efetivo depois que a instrução no slot de retardo é executada. Se o desvio não for obtido, a instrução no slot de atraso não será executada.
BC1T		Branch on FP False Likely	BC1T cc, offset	<i>if FPConditionCode(cc) = 1 then branch</i>	Testa um código de condição FP e faz um desvio condicional relativo ao PC. Se o bit de código de condição FP cc for verdadeiro (1), o programa se ramifica para o endereço de destino efetivo após a instrução no slot de atraso
BC1TL		Branch on FP True Likely	BC1TL cc, offset	<i>if FPConditionCode(cc) = 1 then branch_likely</i>	Testa um código de condição FP e faz um desvio condicional relativo ao PC. Execute a instrução no slot de atraso somente se o desvio for tomado. Se o bit cc do código de condição FP for verdadeiro (1), o programa desvia para o endereço de destino efetivo depois que a instrução no slot de retardo é executada. Se o desvio não for obtido, a instrução no slot de atraso não será executada.

INSTRUÇÃO	CÓDIGO	NOME	FORMATO	OPERAÇÃO	DESCRIÇÃO DETALHADA
BC2EQZ		Branch if Coprocessor 2 Condition (Register) is Equal to Zero	BC2EQZ ct, offset	<i>if COP2Condition[ct] = 0 then branch</i>	Desvia se a condição do coprocessador 2 (registrador) for igual OU diferente de zero. Desvia se a condição do coprocessador 2 for igual a zero. Se a condição do coprocessador 2 for verdadeira, o desvio é tomado.
BC2NEZ		Branch if Coprocessor 2 Condition (Register) is Not Equal to Zero	BC2NEZ ct, offset	<i>if COP2Condition[ct] != 0 then branch</i>	Desvia se a condição do coprocessador 2 for igual OU diferente de zero. Desvia se a condição do coprocessador 2 for diferente de zero. Se a condição do coprocessador 2 for falsa, o desvio é tomado
BC2F		Branch on COP2 False	BC2F cc, offset	<i>if COP2Condition(cc) = 0 then branch</i>	Testa um código de condição COP2 e faz um desvio condicional relativo ao PC. Se a condição COP2 especificada por cc for falsa (0), o programa desvia para o endereço de destino efetivo depois que a instrução no slot de atraso é executada.
BC2FL		Branch on COP2 False Likely	BC2FL cc, offset	<i>if COP2Condition(cc) = 0 then branch_likely</i>	Testa um código de condição COP2 e faz um desvio condicional relativo ao PC. Executa a instrução no slot de atraso somente se o desvio for tomado. Se a condição COP2 especificada por cc for falsa (0), o programa desvia para o endereço de destino efetivo depois que a instrução no slot de atraso é executada. Se o desvio não for obtido, a instrução no slot de atraso não será executada.
BC2T		Branch on COP2 True	BC2T cc, offset	<i>if COP2Condition(cc) = 1 then branch</i>	Testa um código de condição COP2 e faz um desvio condicional relativo ao PC. Se a condição COP2 especificada por cc for verdadeira (1), o programa desvia para o endereço de destino efetivo depois que a instrução no intervalo de atraso é executada.
BC2TL		Branch on COP2 True Likely	BC2TL cc, offset	<i>if COP2Condition(cc) = 1 then branch_likely</i>	Testa um código de condição COP2 e faz um desvio condicional relativo ao PC. Execute a instrução no slot de atraso somente se o desvio for tomado. Se a condição COP2 especificada por cc for verdadeira (1), o programa desvia para o endereço de destino efetivo depois que a instrução no intervalo de atraso é executada. Se o desvio não for obtido, a instrução no slot de atraso não será executada
BEQ		Branch on Equal	BEQ rs, rt, offset	<i>if RS = GPR[rt] then branch</i>	Compara os registradores e faz um desvio condicional relativo ao PC. Se os conteúdos de RS e RT forem iguais, desvie para o endereço de destino efetivo depois que a instrução no slot de atraso for executada.
BEQL		Branch on Equal Likely	BEQL rs, rt, offset	<i>if RS = GPR[rt] then branch_likely</i>	Comparar registradores e faz um desvio condicional relativo ao PC. Executa o slot de atraso somente se o desvio for tomado. Se o conteúdo de RS e RT for igual, desvia para o endereço de destino depois que a instrução no intervalo de atraso é executada. Se o desvio não for obtido, a instrução no slot de atraso não será executada.

INSTRUÇÃO	CÓDIGO	NOME	FORMATO	OPERAÇÃO	DESCRIÇÃO DETALHADA
BGEZ		Branch on Greater Than or Equal to Zero	BGEZ rs, offset	<i>if RS >= 0 then branch</i>	Testa um registrador e faz um desvio condicional relativo ao PC. Se o conteúdo de RS for maior ou igual a zero (o bit de sinal é 0), desvia para o endereço de destino efetivo depois que a instrução no slot de retardo é executada. Desvio compactos não têm slots de atraso. A instrução após um desvio compactado só é executada se o desvio não for tomado.
BGEZAL		Branch on Greater Than or Equal to Zero and Link	BGEZAL rs, offset	<i>if RS >= 0 then procedure_call</i>	Testa um registrador e faz uma chamada de procedimento condicional relativo ao PC. Se o conteúdo de RS for maior ou igual a zero (o bit de sinal é 0), desvia para o endereço de destino efetivo depois que a instrução no slot de retardo é executada. Desvio compactos não têm slots de atraso. A instrução após um desvio compactado só é executada se o desvio não for tomado.
BLEZALC		Branch on Less – Compact Zero-Compare and Branch-and-Link	BLEZALC rt, offset	<i>if condition(RT) then procedure_call branch (no delay slot)</i>	Desvia e liga compactado se RT for menor ou igual a zero. Se a condição for verdadeira, o desvio é tomado. A condição é verdadeira se e somente se RT for menor ou igual a zero. Desvio compactos não têm slots de atraso. A instrução após um desvio compactado só é executada se o desvio não for tomado.
BGEZALC		Branch on Greather – Compact Zero-Compare and Branch-and-Link	BLEZALC rt, offset	<i>if condition(RT) then procedure_call branch (no delay slot)</i>	Desvia e liga compactado se RT for maior ou igual a zero. Se a condição for verdadeira, o desvio é tomado. A condição é verdadeira se e somente se RT for maior ou igual a zero. Desvio compactos não têm slots de atraso. A instrução após um desvio compactado só é executada se o desvio não for tomado.
BGTZALC		Branch on Greather then zero – Compact Zero-Compare and Branch-and-Link	BLEZALC rt, offset	<i>if condition(RT) then procedure_call branch (no delay slot)</i>	Desvia e liga compactado se RT for maior que zero. Se a condição for verdadeira, o desvio é tomado. A condição é verdadeira se e somente se RT for maior que zero. Desvio compactos não têm slots de atraso. A instrução após um desvio compactado só é executada se o desvio não for tomado.
BLTZALC		Branch on Less then zero – Compact Zero-Compare and Branch-and-Link	BLEZALC rt, offset	<i>if condition(RT) then procedure_call branch (no delay slot)</i>	Desvia e liga compactado se RT for menor que zero. Se a condição for verdadeira, o desvio é tomado. A condição é verdadeira se e somente se RT for maior que zero. Desvio compactos não têm slots de atraso. A instrução após um desvio compactado só é executada se o desvio não for tomado.

INSTRUÇÃO	CÓDIGO	NOME	FORMATO	OPERAÇÃO	DESCRIÇÃO DETALHADA
BEQZALC		Branch on Equal – Compact Zero-Compare and Branch-and-Link	BLEZALC rt, offset	<i>if condition(RT) then procedure_call branch (no delay slot)</i>	Desvia e liga compactado se RT for igual a zero. Se a condição for verdadeira, o desvio é tomado. A condição é verdadeira se e somente se RT for igual a zero. Desvio compactos não têm slots de atraso. A instrução após um desvio compactado só é executada se o desvio não for tomado.
BNEZALC		Branch on not Equal – Compact Zero-Compare and Branch-and-Link	BLEZALC rt, offset	<i>if condition(RT) then procedure_call branch (no delay slot)</i>	Desvia e liga compactado se RT não for igual a zero. Se a condição for verdadeira, o desvio é tomado. A condição é verdadeira se e somente se RT não for igual a zero. Desvio compactos não têm slots de atraso. A instrução após um desvio compactado só é executada se o desvio não for tomado.
BGEZALL		Branch on Greater Than or Equal to Zero and Link Likely	BGEZALL rs, offset	<i>If RS >= 0 then procedure_call_likely</i>	Testar um registrador e faz uma chamada de procedimento condicional relativa ao PC. Executa o slot de atraso somente se o desvio for tomado. Se o conteúdo de RS for maior ou igual a zero (o bit de sinal é 0), desvia para o endereço de destino efetivo depois que a instrução no slot de retardo é executada. Se o desvio não for obtido, a instrução no slot de atraso não será executada.
B<cond>C		Compact Compare-and-Branch Instructions		<i>if condition(RS and/or R) then compact branch (no delay slot)</i>	
BEQC		Branch on Equal Compact Compare	BEQC rs, rt, offset	<i>if condition(RS and/or R) then compact branch (no delay slot)</i>	Desvio compacto se RS e RT são iguais. Compara igualdade e desigualdade entre registradores e desvia com deslocamento de 16 bits
BNEC		Branch on not Equal Compact Compare	BNEC rs, rt, offset	<i>if condition(RS and/or R) then compact branch (no delay slot)</i>	Desvio compacto se RS e RT são diferentes. Compara igualdade/desigualdade entre registradores e desvia com deslocamento de 16 bits
BLTC			BLTC rs, rt, offset	<i>if condition(RS and/or R) then compact branch (no delay slot)</i>	Desvio compacto se RS é menor que RT. Compara registradores com sinal e desvia com deslocamento de 16 bits
BGEC			BGEC rs, rt, offset	<i>if condition(GPR[rs] and/or GPR[rt]) then compact branch (no delay slot)</i>	Desvio compacto se RS é maior ou igual que RT. Compara registradores com sinal e desvia com deslocamento de 16 bits
BLTUC			BLTUC rs, rt, offset	<i>if condition(RS and/or R) then compact branch (no delay slot)</i>	Desvio compacto se RS é menor que RT sem sinal. Compara registrador-registrador sem sinal e desvia com deslocamento de 16 bits
BGEUC			BGEUC rs, rt, offset	<i>if condition(RS and/or R) then compact branch (no delay slot)</i>	Desvio Compacto se RS é maior ou igual a RT se sinal. Compara registrador-registrador sem sinal e desvia com deslocamento de 16 bits
BGTC			BGTC rs, rt, offset	<i>if condition(RS and/or R) then compact branch (no delay slot)</i>	Desvio compacto se RT é maior que RS (alias for BLTC). Idiomas Assembly com operandos invertidos para comparação e desvio com e sem sinal
BLEC			BLEC rs, rt, offset	<i>if condition(RS and/or R) then compact branch (no delay slot)</i>	Desvio compacto se RT é menor ou igual a RS (alias for BGEC). Idiomas Assembly com operandos invertidos para comparação e desvio com e sem sinal

INSTRUÇÃO	CÓDIGO	NOME	FORMATO	OPERAÇÃO	DESCRIÇÃO DETALHADA
BGTUC			BGTUC rs, rt, offset	<i>if condition(RS and/or R) then compact branch (no delay slot)</i>	Desvio compacto se RT é maior que RS sem sinal (alias for BLTUC). Idiomas Assembly com operandos invertidos para comparação e desvio com e sem sinal
BLEUC			BLEUC rs, rt, offset	<i>if condition(RS and/or R) then compact branch (no delay slot)</i>	Desvio compacto se RT é menor ou igual a RT sem sinal (alias for BGEUC). Idiomas Assembly com operandos invertidos para comparação e desvio com e sem sinal
BLTZC			BLTZC rt, offset	<i>if condition(RS and/or R) then compact branch (no delay slot)</i>	Desvio compacto se RT é menor que zero. Compara um Registrador com sinal com o valor zero e desvia com deslocamento de 16 bits
BLEZC			BLEZC rt, offset	<i>if condition(RS and/or R) then compact branch (no delay slot)</i>	Desvio compacto se RT é menor ou igual a zero. Compara um Registrador com sinal com o valor zero e desvia com deslocamento de 16 bits
BGEZC			BGEZC rt, offset	<i>if condition(RS and/or R) then compact branch (no delay slot)</i>	Desvio compacto se RT é maior ou igual a zero. Compara um Registrador com sinal com o valor zero e desvia com deslocamento de 16 bits
BGTZC			BGTZC rt, offset	<i>if condition(RS and/or R) then compact branch (no delay slot)</i>	Desvio compacto se RT é maior que zero. Compara um Registrador com sinal com o valor zero e desvia com deslocamento de 16 bits
BEQZC			BEQZC rs, offset	<i>if condition(RS and/or R) then compact branch (no delay slot)</i>	Desvio compacto se RS é igual a zero. Compara registrador igual/não igual com zero e desvia com deslocamento de 21 bits
BNEZC			BNEZC rs, offset	<i>if condition(RS and/or R) then compact branch (no delay slot)</i>	Desvio compacto se RS não é igual a zero. Compara registrador igual/não igual com zero e desvia com deslocamento de 21 bits
BGEZL		Branch on Greater Than or Equal to Zero Likely	BGEZL rs, offset	<i>if RS >= 0 then branch_likely</i>	Testa um registrador e faz um desvio condicional relativo ao PC. Executa o slot de atraso somente se o desvio for tomado. Se o conteúdo de RS for maior ou igual a zero (o bit de sinal é 0), desvia para o endereço de destino efetivo depois que a instrução no slot de retardo é executada. Se o desvio não for tomado, a instrução no slot de atraso não será executada.
BGTZ		Branch on Greater Than Zero	BGTZ rs, offset	<i>if RS > 0 then branch</i>	Testa um registrador e faz um desvio condicional relativo ao PC. Se o conteúdo de RS for maior que zero (o bit de sinal é 0, mas o valor não é zero), avance para o endereço de destino efetivo após a execução da instrução no slot de retardo.
BGTZL		Branch on Greater Than Zero Likely	BGTZL rs, offset	<i>if RS > 0 then branch_likely</i>	Testa um registrador e faz um desvio condicional relativo ao PC. Executa o slot de atraso somente se o desvio for tomado. Se o conteúdo de RS for maior que zero (o bit de sinal é 0, mas o valor não é zero), desvie para o endereço de destino efetivo após a execução da instrução no slot de retardo. Se o desvio não for tomado, a instrução no slot de atraso não será executada

INSTRUÇÃO	CÓDIGO	NOME	FORMATO	OPERAÇÃO	DESCRIÇÃO DETALHADA
BITSWAP		Swaps (reverses) bits in each byte	BITSWAP rd,rt	$RD.byte(i) = reverse_bits_in_byte(RT.byte(i)), \text{ for all bytes } i$	Cada byte na entrada RT é movido para a mesma posição de byte na saída RD, com bits em cada byte invertidos. O BITSWAP opera em todos os 4 bytes de um registrador de 32 bits em uma CPU de 32 bits.
BLEZ		Branch on Less Than or Equal to Zero	BLEZ rs, offset	$if RS \leq 0 \text{ then branch}$	Testa um registrador e faz um desvio condicional relativo ao PC. Se o conteúdo de RS for menor ou igual a zero (o bit de sinal é 1 ou o valor é zero), desvia para o endereço de destino efetivo depois que a instrução no slot de atraso é executada.
BLEZL		Branch on Less Than or Equal to Zero Likely	BLEZL rs, offset	$if RS \leq 0 \text{ then branch_likely}$	Testa um registrador e faz um desvio condicional relativo ao PC. Executa o slot de atraso somente se o desvio for tomado. Se o conteúdo de RS for menor ou igual a zero (o bit de sinal é 1 ou o valor é zero), desvia para o endereço de destino efetivo depois que a instrução no slot de atraso é executada. Se o desvio não for tomado, a instrução no slot de atraso não será executada.
BLTZ		Branch on Less Than Zero	BLTZ rs, offset	$if RS < 0 \text{ then branch}$	Testa um registrador e faz um desvio condicional relativo ao PC. Se o conteúdo de RS for menor que zero (o bit de sinal é 1), desvia para o endereço de destino efetivo depois que a instrução no slot de retardo é executada.
BLTZAL		Branch on Less Than Zero and Link	BLTZAL rs, offset	$if RS < 0 \text{ then procedure_call}$	Testa registrador e faz uma chamada de procedimento condicional relativa ao PC. Se o conteúdo de RS for menor que zero (o bit de sinal é 1), desvia para o endereço de destino efetivo depois que a instrução no slot de retardo é executada.
BLTZALL		Branch on Less Than Zero and Link Likely	BLTZALL rs, offset	$if RS < 0 \text{ then procedure_call_likely}$	Testa um registrador e faz uma chamada de procedimento condicional relativa ao PC. Executa o slot de atraso somente se o desvio for tomado. Se o conteúdo de RS for menor que zero (o bit de sinal é 1), desvia para o endereço de destino efetivo depois que a instrução no slot de retardo é executada. Se o desvio não for tomado, a instrução no slot de atraso não será executada.
BLTZL		Branch on Less Than Zero Likely	BLTZL rs, offset	$if RS < 0 \text{ then branch_likely}$	Testa um registrador e faz um desvio condicional relativo ao PC. Executa o slot de atraso somente se o desvio for tomado. Se o conteúdo de RS for menor que zero (o bit de sinal é 1), desvia para o endereço de destino efetivo depois que a instrução no slot de retardo é executada. Se o desvio não for tomado, a instrução no slot de atraso não será executada.
BNE		Branch on not Equal	BNE rs, rt, offset	$if RS \neq RT \text{ then branch}$	Comparar registradores e faz um desvio condicional relativo ao PC. Se os conteúdos de RS e RT não forem iguais, desvia para o endereço de destino efetivo após a instrução no slot de retardo ser executada.

INSTRUÇÃO	CÓDIGO	NOME	FORMATO	OPERAÇÃO	DESCRIÇÃO DETALHADA
BNEL		Branch on Not Equal Likely	BNEL rs, rt, offset	<i>if RS != RT then branch_likely</i>	Compara registradores e faz um desvio condicional relativo ao PC. Executa o slot de atraso somente se o desvio for tomado. Se os conteúdos de RS e RT não forem iguais, desvia para o endereço de destino efetivo depois que a instrução no slot de atraso for executada. Se o desvio não for tomado, a instrução no slot de atraso não será executada
BOVC		Branch on Overflow, Compact; Branch on No Overflow, Compact	BOVC rs,rt,offset	<i>branch if/if-not NotWordValue(RS + RT)</i>	Detecta overflow para adição com sinal em 32 bits e desvia se há o overflow.
BNVC		Branch on Overflow, Compact; Branch on No Overflow, Compact	BNVC rs,rt,offset	<i>branch if/if-not NotWordValue(RS + RT)</i>	Detecta overflow para adição com sinal em 32 bits e desvia se não há o overflow.