

### OBSERVAÇÕES:

1. Na sintaxe do assembler descrito, em instruções de dois operandos, a seguinte convenção deve ser respeitada:

Ex: sub  $R_i, R_j$ ; onde:  $R_i$  é o 1º operando ou operando fonte ( $0 \leq i \leq 4$ )

$R_j$  é o 2º operando ou operando destino ( $0 \leq j \leq 4$ )

No caso, a operação executada é a subtração  $\langle R_j - R_i \rangle$  e o resultado é armazenado na posição do destino, ou seja, carregado em  $R_j$ . O operando destino é perdido neste caso. Em se tratando da instrução de comparação, esta deverá ter sempre como referência o operando fonte, ou seja, o sentido da comparação é sempre o operando fonte sendo comparado com o valor do operando destino. Se o operando fonte for maior que o destino, então o desvio que se segue (brg p/ exemplo, será tomado). A instrução cmp é na verdade uma subtração  $\langle R_j - R_i \rangle$ .

- 2) O simulador deverá aceitar constantes numéricas positivas e negativas como operandos imediatos expressas tanto em decimal como hexadecimal. Recomenda-se representar internamente os valores negativos em "complemento a 2".
- 3) Recomenda-se partir da arquitetura básica /ilustrada em aula e ir adicionando blocos (registros, multiplexadores, decodificadores, etc) a medida que forem sendo necessários.
- 4) Implementações adicionais tais como instruções extras, modos de endereçamento mais elaborados, estrutura de stack, unidades funcionais especializadas, memória cache, etc, embora opcionais, serão levados em consideração para efeitos da nota final do trabalho. A nota de cada grupo de trabalho só será dada após a apresentação do último grupo.