1. Na implementação Multiciclo, cada etapa da execução levara 1 ciclo de Clock, onde o tamanho do Clock é normalizado de acordo com a etapa mais demorada. Isso permite que mais de um componente seja usado durante a execução de uma instrução. Isso pode reduzir consideravelmente a quantidade de componentes, fazendo com que o preço do mesmo seja menor e sua performance maior, em relação ao Uniciclo.
2. O processamento da instrução deve ser dividido em 5 estágio, referentes as etapas de execução do Multiciclo. Os estados desses estágios são guardados em registradores intermediários para que haja capacidade de implementação da técnica de *by-pass.*
4. Dependência RAW: O valor do registrador F1 é requisitado na 2º Instrução, quando a 1º Instrução não terminou de ser executada;

Dependência RAW: O valor do registrador F4 é requisitado na 3º Instrução, quando a 2º Instrução não terminou de ser executada;

Dependência RAW: O valor do registrador F5 é requisitado na 5º Instrução, quando a 4º Instrução não terminou de ser executada;

Dependência WAW: O valor do registrador F1 é requisitado na 2º Instrução, quando a 1º Instrução não terminou de escrever no mesmo;

Dependência WAR: É criado um *Stall* durante o escalonamento das instruções fazendo com que a 3º Instrução se torne dependente da 2º Instrução, em razão do valor de F1.

1. CPU tempo ideal = (CPU ciclos de clock + Ciclos de Clock de Stall de Memória) x Período de ciclo de Clock =>

CPU tempo ideal = (IC x CPI + 0) x Período de ciclo de Clock =>

CPU tempo ideal = IC x 1 x Período de ciclo de Clock

Ciclos de Clock de *Stall* de memória = IC x (1 + 0,5) x 0.02 x 25 = IC x 0,75

CPU tempo real = (IC x 1 + IC x 0,75) x Período de ciclo de Clock =>

CPU tempo real = IC x 1,75 x Período de ciclo de Clock

Comparando os desempenhos com *Stalls* pelo ideal:

a) Escrita ao mesmo tempo no cache e na memória.

b) Escrita somente no cache e atualização na memória somente na substituição do bloco.

c) Se valendo dos princípios de localidade, existe o conceito de localidade temporal, onde se um item é referenciado, ele tende a ser referenciado novamente dentro de um espaço de tempo curto.

d) Diferentemente da anterior, este leva em consideração o espaço, e supõe que: se um item foi referenciado recentemente, logo itens adjacentes, provavelmente, serão referenciados.