



1. Considere a arquitetura MIPS Ciclo Único com Pipeline, onde seja possível suportar as instruções detalhadas embaixo.

Código	Registros	Memória
<p>Apresentar um Código principal que chame uma função capaz de resolver a série de Fibonacci para cada valor armazenado em memória.</p> <p>O resultado obtido, será armazenado no mesmo endereço de memória onde o número utilizado para executar a série foi obtido.</p> <p>Isto será realizado para todos os endereços listados nessa tabela.</p> <p>Observações:</p> <ul style="list-style-type: none"> Incluir todos os tipos de hazards na implementação pensando na sua execução em Microarquitetura Single Cycle com Pipeline. Incluir endereços para cada instrução. Desta forma serão identificados de acordo com o comportamento da memória de instruções. 		
	\$0 00 00 00 00	00 00 00 2c 01 ac 1b 00
	...	00 00 00 28 aa 71 bb f0
	\$t0 00 00 00 00	00 00 00 24 ff 0c 0e ee
	\$t1 00 00 00 00	00 00 00 20 00 00 fe 00
	\$t2 00 00 00 08	00 00 00 1c a2 5f 44 00
	...	00 00 00 18 01 22 44 59
	\$s0 00 00 00 00	00 00 00 14 00 00 00 00
	\$s1 00 00 00 00	00 00 00 10 bb 00 00 00
	\$s2 00 00 00 00	00 00 00 0c 00 00 00 10
	\$s3 00 00 00 00	00 00 00 08 10 fe dc ba
	\$s4 00 00 00 00	00 00 00 04 00 00 00 00
	\$s5 00 00 00 00	00 00 00 00 13 57 9b cd

- Apresentar os registros e a memória após a execução dos códigos listados previamente. Apresentar esse código em MARS.
- Desenhar o caminho de dados com hazards (Forwarding e Stalls) existentes, elementos dos hazards, e sistemas de controle. Evidenciar os sinais de controle e sistema de controle do hazard necessários para cada uma das instruções listadas. Apresentar as equações implementadas no sistema de controle de hazard para realizar a toma de decisão.
- Evidenciar comportamento do sistema proposto com hazards existentes. DIAGRAMA DE BLOCOS.
- Implementar um processador superescalar para o processador pipeline previamente detalhado.
- Reordene o código de forma que produza o mesmo resultado, de forma a eliminar a ocorrência de Hazards para o processador pipeline e para o processador superescalar.
- Implementar um processador multiciclo, e single cycle, que suporte as mesmas instruções previamente apresentadas e comparar o desempenho dos três tipos de processadores após executar 5 Milhões de vezes o conjunto de instruções obtidas nas seções anteriores, tendo em conta os tempos apresentados a seguir:

Elemento	Parâmetro	Atraso (ps)
Reg clk to Q	t_{pcqpc}	25
Reg setup	t_{setup}	10
Multiplexer	t_{mux}	15
ALU	t_{ALU}	120
Mem. read	t_{mem}	200
Reg. file read	t_{RFread}	70
Reg. file setup	$t_{RFsetup}$	5

2. Concatenar com o processador pipeline previamente desenhado um sistema de memória física e um disco rígido. Desenhar o caminho de dados.

- As interações com memória geradas na sequência de instruções detalhadas no ponto anterior serão implementadas em uma CACHE com mapeamento direto, em uma associativa de n-vias e em uma totalmente associativa. As CACHES L1 e L2 estão dentro do processador com taxa de erro de 30% e tamanho de 64KB para L1, e taxa de erro de 15% e tamanho de 512KB para L2. L3 está fora do processador com taxa de erro de 5% e tamanho de 2 MB, e a Memória principal tem um tamanho de 8GB. Desenhar o hardware e as tabelas para cada um dos tipos de CACHE previamente mencionados, incluindo a correlação entre as CACHES. Apresentar mapa de conjuntos para cada tipo de CACHE (Número de conjuntos, bits necessários para mapear na memória principal o conjunto no qual pertence cada endereço, e número de bits correspondentes ao tag_bit).
- Comparar o miss rate obtido por cada um dos tipos de CACHE avaliados e o tempo médio de acesso à memória. Justificar caso seja evidenciada alguma diferença entre os tipos de CACHES avaliadas.
- A memória virtual possui um tamanho de 15GB, a memória física tem um tamanho de 8GB e o tamanho da página é 4KB. Desenhar as páginas da memória virtual e da memória física. Realizar para uma memória virtual de um e dois níveis.
- Evidenciar o esquema necessário para converter um endereço virtual em um endereço físico.