

GS1013 – Arq. e Org. de Computadores

Organização MIPS32 Monociclo

Arquitetura e Organização de Computadores

Universidade Federal de Uberlândia
Faculdade de Computação
Prof. Dr. rer. nat. Daniel D. Abdala

Nesta Aula

- Formatos de Instrução;
- Subconjunto de Instruções Implementadas;
- Revisão sobre Memórias;
- Organização Harvard de Memória;
- Busca de Instruções;
- Registradores e banco de Registradores;
- Projeto de uma ULA simples;
- Visão detalhada do projeto de circuitos lógicos;
- Visão detalhada do projeto de circuitos aritméticos;
- Caminho de Dados: Instruções Tipo-R.

Projeto do Subset ISA-MIPS Monociclo

- Hardware projetado para concordar com a ISA;
- Processador pode ser subdividido nas seguintes unidades funcionais:

Prof. Dr. rer. nat. Daniel Duarte Abdala

MIPS-Monociclo

- Significa que uma instrução é executada por ciclo de clock.
- Instruções distintas requerem tempos mínimos que variam.
- O Ciclo de clock deve ser definido para acomodar a instrução mais lenta;
- Não é usada atualmente pois é ineficiente;

Prof. Dr. rer. nat. Daniel Duarte Abdala

Metodologia de Clock

- Define quando sinais podem ser lidos ou escritos;
- Geralmente utiliza-se uma metodologia gatilhada por mudança de estado do clock;

Prof. Dr. rer. nat. Daniel Duarte Abdala

Formatos de Instruções

Prof. Dr. rer. nat. Daniel Duarte Abdala

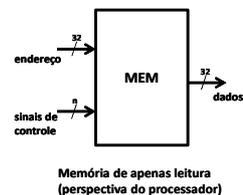
Subconjunto de Instruções

Instrução	Exemplo	Semântica	Tipo
add	add \$rd,\$rs,\$rt	$rd \leftarrow rs + rt$	R
sub	sub \$rd,\$rs,\$rt	$rd \leftarrow rs - rt$	R
and	and \$rd,\$rs,\$rt	$rd \leftarrow rs \& rt$	R
or	or \$rd,\$rs,\$rt	$rd \leftarrow rs rt$	R
nor	nor \$rd,\$rs,\$rt	$rd \leftarrow \neg(rs rt)$	R
addi	addi \$rd,\$rs,\$cte	$rd \leftarrow rs + cte$	I
andi	andi \$rd,\$rs,\$cte	$rd \leftarrow rs \& cte$ (16 bits lab)	I
ori	ori \$rd,\$rs,\$cte	$rd \leftarrow rs cte$ (16 bits lab)	I
andi	andi \$rd,\$rs,\$cte	$rd \leftarrow \neg(rs + cte)$ (16 bits lab)	I
beq	beq \$rs,\$rt,LABEL	SE $rs=rt$ pule para LABEL	R
slt	slt \$rd,\$rs,\$rt	SE $rs < rt$ SETA $rd=01$ SENÃO $rd=00$	R
slti	slti \$rd,\$rs,\$cte	SE $rs < cte$ SETA $rd=01$ SENÃO $rd=00$	I
lw	lw \$rt,\$cte(\$rs)	$rt \leftarrow MEM[rs+cte]$	I
sw	sw \$rs,\$cte(\$rs)	$MEM[rs+cte] \leftarrow rs$	I
j	j LABEL	pule para LABEL	J

7

Memória de Programa

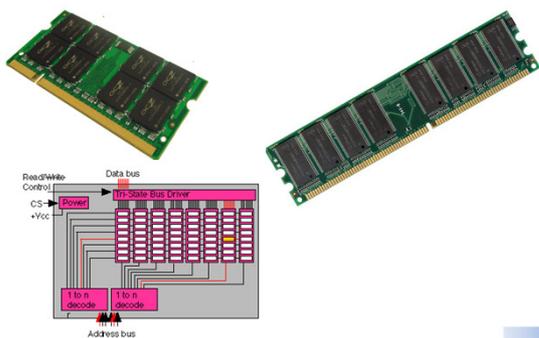
- 2^{30} endereços
- 4 bytes por instrução
- 1.073.741.824 bytes
- 268.435.456 instruções
- “texto” do programa deve ser alterado apenas pelo SO;
- Arquitetura Harvard – Programas e dados são armazenados em memórias distintas.



Prof. Dr. rer. nat. Daniel Duarte Abdala

8

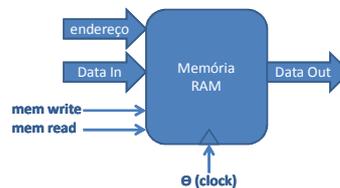
Memória RAM



9

Memória RAM

- Memória principal do sistema;
- Contém dados e programas;
- Volátil;
- Lenta em comparação aos registradores;



Abstração Memória:

10

Abstração de Memória

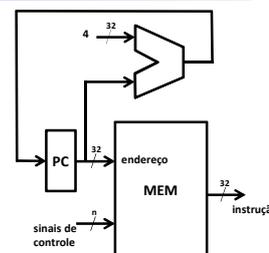


- Do ponto de vista do programador:
 - Array de dados;
 - Endereço identifica uma célula de memória individual;
 - O conteúdo da célula contém efetivamente o dado/instrução;

11

Busca de Instruções

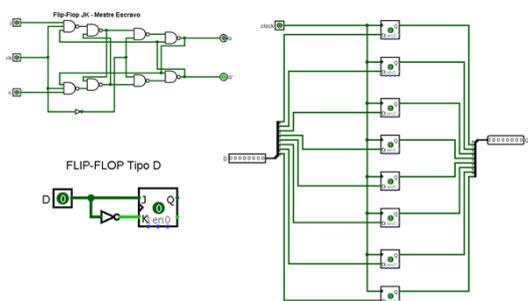
- Qual a prox. instrução?
- Somam-se 4 bytes pois cada instrução na ISA considerada tem 4 bytes;
- PC sempre aponta para a próxima instrução a ser executada;
- IR contém a instrução sendo executada.



Prof. Dr. rer. nat. Daniel Duarte Abdala

12

Revisão - Registrador



13

Registradores

- Apenas 32 registradores?
 - Poucos e rápidos!
- E os registradores \$at (1) e \$k0,\$k1 (26,27)?
 - Montador e SO

Nome	# de registrador	Uso	Preservado em Chaveado?
\$zero	0	Constante 0	
\$ra	2,3	Retorno de funções	Não
\$a0-\$a3	4-7	Argumentos	Não
\$t0-\$t7	8-15	Temporários	Não
\$s0-\$s7	16-23	Salvos	Sim
\$t8-\$t9	24,25	Mais temporários	Não
\$gp	28	Ponteiro global	Sim
\$sp	29	Ponteiro de pilha	Sim
\$fp	30	Ponteiro de quadro	Sim
\$ra	31	Endereço de retorno	Sim

14

Banco de Registradores

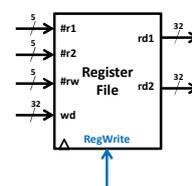
- Register File;
- Parte da CPU que implementa os registradores acessíveis;
- Sempre acessa dois registradores por vez;
- Uma vez endereçado o registrador nas entradas "#r1" e "#r2" o conteúdo dos registradores correspondentes é imediatamente direcionado para as saídas "rd1" e "rd2";

Prof. Dr. rer. nat. Daniel Duarte Abdala

15

Register File

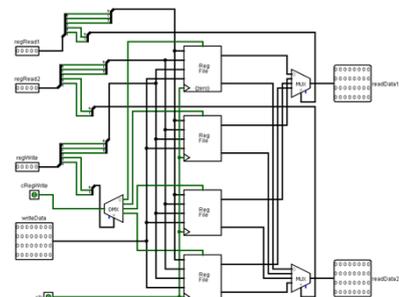
- O File Register prevê a escrita de um registrador por ciclo de clock.
- O endereço do registrador deve ser especificado no campo "#rw" e o dado a ser escrito no campo "wd";
- O sinal de controle "RegWrite" deve ser colocado em "1" para que a escrita tenha efeito.



Prof. Dr. rer. nat. Daniel Duarte Abdala

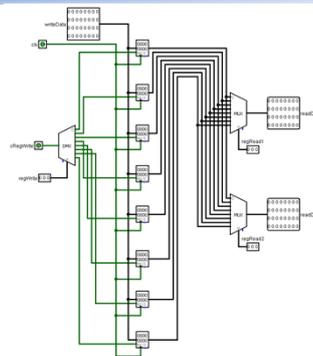
16

Implementação Lógica do RF



17

Banco de 8 Registradores



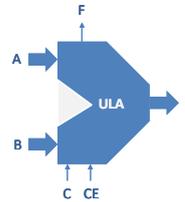
18

ULA – Unidade Lógica e Aritmética

- “Coração” do μ processador;
- É a coleção de subsistemas digitais que executam boa parte das funções de um processador;

19

ULA – Visão Geral



- **A,B** – entradas de dados;
- **C** – controle, indica que operação será executada;
- **S** – saída de dados;
- **F** – flags, indica o estado da ULA;
- **CE** – chip enable, habilita o funcionamento da ULA.

20

Operações Contempladas

Operações

1. Adição
2. Subtração
3. E lógico
4. OU lógico
5. NÃO-OU lógico
6. XOU lógico
7. Comparação*

Considerações

- Projetar um subsistema para cada operação;
- Executar todas as operações sempre;
- Selecionar o resultado que se deseja via multiplexação;
- Resultados das comparações são espelhados diretamente nos flags.

21

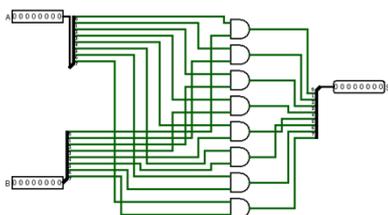
Flags

me lg ma io iu

- Sinais (bits) que indicam o estado da ULA;
 1. Me $\rightarrow A < B$
 2. Ig $\rightarrow A = B$
 3. Ma $\rightarrow A > B$
 4. Io \rightarrow Integer Overflow
 5. Iu \rightarrow Integer Underflow

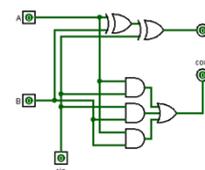
22

Subsistema “E” (8 bits)



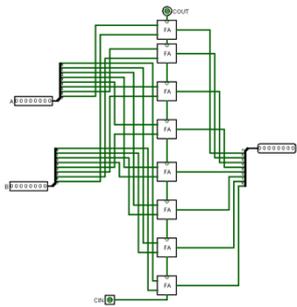
23

Somador Completo (1 bit)



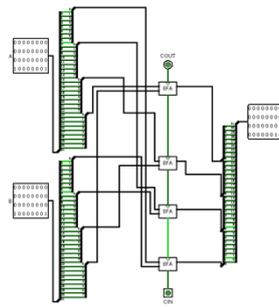
24

Somador (8 bits)



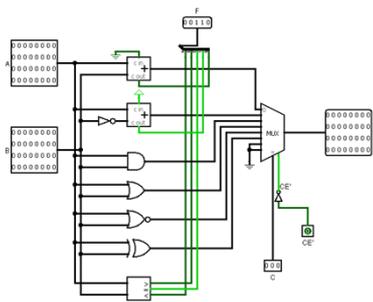
25

Somador (32 bits)



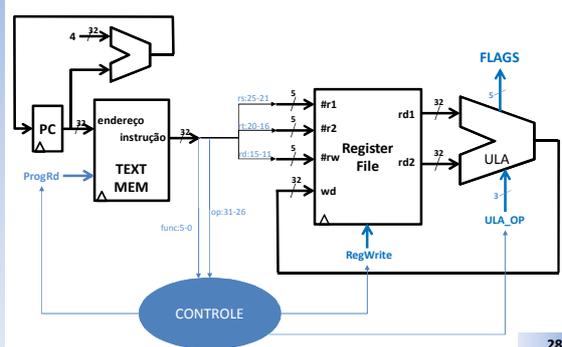
26

ULA – Diagrama Esquemático



27

Caminho de Dados – Apenas Tipo R



28