



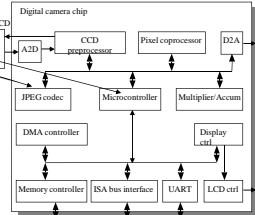
Aula 2

Engenharia de Sistemas Embarcados
 Prof. Abel Guilhermino
 Tópico: Projeto de Sistemas Embarcados




Introdução

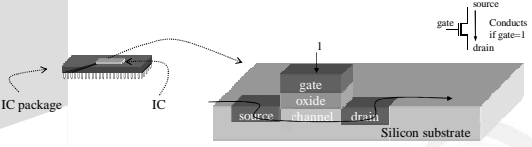
- Processador
 - Circuito digital que realiza tarefas computacionais
 - Controlador e caminho de dados
 - Propósito geral: variedade tarefas computacionais
 - Uso específico: realiza uma tarefa em particular
 - Customizado: tarefa não padrão
- Um processador de propósito específico customizado pode ser:
 - Rápido, pequeno e de baixo consumo
 - Mas, alto custo NRE, time-to-market longo, pouco flexível



Engenharia de Sistemas Embarcados
 Grupo de Engenharia da Computação - CIn / UFPE

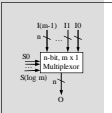
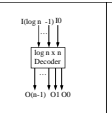
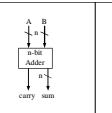
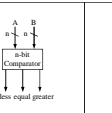
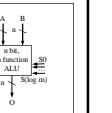
CMOS transistor on silicon

- Transistor
 - O componente elétrico básico em sistemas digitais
 - Age como uma chave liga/desliga
 - Tensão no "gate" controla o fluxo de corrente da fonte para o dreno
 - Não confunda "gate" com "logic gate"



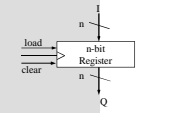
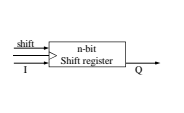
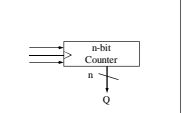
Engenharia de Sistemas Embarcados
 Grupo de Engenharia da Computação - CIn / UFPE

Componentes combinacionais

				
<p>0 = 00 if S=0.00 1 = 01 if S=0.01 ... i(m-1) = 1 if S=1.11</p>	<p>00 = 1 if I=0.00 01 = 1 if I=0.01 ... 0(n-1) = 1 if I=1.11</p>	<p>sum = A + B (first n bits) carry = (n+1)th bit of A+B</p>	<p>less = 1 if A < B equal = 1 if A = B greater = 1 if A > B</p>	<p>O = A op B op determined by S. O = 0 if S=1.11</p>
	<p>With enable input e → all O's are 0 if e=0</p>	<p>With carry-in input Ci → sum = A + B + Ci</p>		<p>May have status outputs carry, zero, etc.</p>

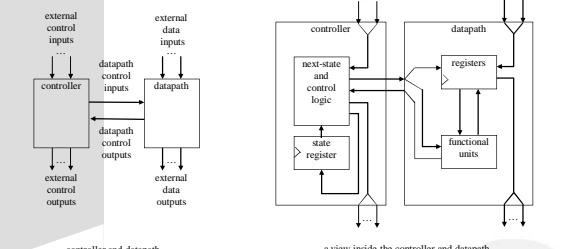
Engenharia de Sistemas Embarcados
 Grupo de Engenharia da Computação - CIn / UFPE

Componentes Sequenciais

		
<p>Q = 0 if clear=1, 1 if load=1 and clock=1, Q(previous) otherwise.</p>	<p>Q = lb - Content shifted - 1 stored in msb</p>	<p>Q = 0 if clear=1, Q(previous)+1 if count=1 and clock=1.</p>

Engenharia de Sistemas Embarcados
 Grupo de Engenharia da Computação - CIn / UFPE

Modelo Básico de Processador Customizado de Propósito Específico



Engenharia de Sistemas Embarcados
 Grupo de Engenharia da Computação - CIn / UFPE

Exemplo: Máximo divisor comum

- Primeiro crie o algoritmo
- Converta o algoritmo para uma máquina de estados "complexa"
 - FSMD: máquina de estados finito com *datapath*
 - Templates podem ser utilizados para a conversão

(a) black-box view

(b) desired functionality

```

0: int x, y;
1: while (1) {
2:   while (!gcd);
3:   x = x_i;
4:   y = y_j;
5:   while (x != y) {
6:     if (x < y)
7:       y = y - x;
8:     else
9:       x = x - y;
10:  }
11:  d_o = x;

```

(c) state diagram

Engenharia de Sistemas Embarcados 7

Criando o fluxo de dados

- Crie um registrador para cada variável declarada
- Crie uma unidade funcional para cada operação aritmética
- Conecte as portas, registradores e unidades funcionais
 - Baseado em leituras e escritas
 - Uso de multiplexadores para múltiplas fontes
- Crie um identificador único
 - Para cada entrada e saída de controle dos componentes do *datapath*

(a) State diagram

(b) Datapath

Engenharia de Sistemas Embarcados 8

Criando o controlador da FSM

- Mesma estrutura da FSMD
- Substitua ações/condições complexas por configurações do *datapath*

(a) State diagram

(b) Controller

(c) Datapath

Engenharia de Sistemas Embarcados 9

Quebrando em um controlador e datapath

(a) Controller implementation model

(b) Datapath

Engenharia de Sistemas Embarcados 10

Processador de Propósito Geral

- Processador de Propósito Geral
 - Processador projetado para uma grande variedade de tarefas computacionais
 - Baixo custo unitário, em parte porque o fabricante divide o custo NRE pelo grande número de unidades
 - Motorola vendeu meio bilhão de microcontroladores 68HC05 no ano de 1996
 - Projetado com bastante cuidado uma vez que o alto custo NRE é aceitável
 - Pode atingir um bom desempenho, tamanho e consumo de potência
 - Time-to-market / protótipo rápido, alta flexibilidade
 - Usuário escreve apenas software; não é feito o projeto do processador

Engenharia de Sistemas Embarcados 11

Arquitetura Básica

- Unidade de Controle de Caminho de dados
 - Similar a processador de propósito específico
- Principais diferenças
 - Caminho de dados genérico
 - Unidade de controle não armazena o algoritmo - o algoritmo é "programado" na memória

Engenharia de Sistemas Embarcados 12

Operações do Datapath

- **Load**
 - Transfere o conteúdo de uma posição de memória para um registrador
- **Operação da ALU**
 - Operando e o resultado de uma operação lógico-aritmética estão armazenados em registradores
- **Store**
 - Transfere o conteúdo de um registrador para uma posição de memória

The diagram shows a processor with a control unit (containing a controller, PC, and IR) and a datapath (containing registers, ALU, and I/O). Memory is connected to the datapath. Arrows indicate the flow of data: from memory to registers, from registers to the ALU, and from the ALU back to registers. The PC and IR are also shown with data flow arrows.

Engenharia de Sistemas Embarcados 13
 Grupo de Engenharia da Computação - CIn / UFPE

Unidade de Controle

- **Unidade de controle:** determina as operações realizadas no datapath
 - A seqüência das operações desejadas ("instruções") são armazenadas na memória - "programa"
- **Ciclo de instrução** - quebra das operações em diversas sub-operações, uma para cada ciclo de relógio, e.g.:
 - **Busca de instrução (Fetch):** lê instrução da memória e armazena no registrador de instruções
 - **Decodificação:** determina o significado da instrução
 - **Busca dos operandos (Fetch operands):** move dados da memória para o registrador do datapath
 - **Execução:** move dados através da ULA
 - **Armazena os resultados:** escreve dados do registrador na memória

The diagram shows the processor control unit and datapath. The control unit contains a controller, PC, and IR. The datapath contains registers, ALU, and I/O. Memory is connected to the datapath. Arrows indicate the flow of data: from memory to registers, from registers to the ALU, and from the ALU back to registers. The PC and IR are also shown with data flow arrows.

Engenharia de Sistemas Embarcados 14
 Grupo de Engenharia da Computação - CIn / UFPE

Sub-operações da unidade de controle

- **Busca**
 - Lê próxima instrução dentro do IR
 - PC (program counter): sempre aponta para a próxima instrução
 - IR: armazena a instrução a ser executada

The diagram shows the processor control unit and datapath. The control unit contains a controller, PC, and IR. The datapath contains registers, ALU, and I/O. Memory is connected to the datapath. Arrows indicate the flow of data: from memory to registers, from registers to the ALU, and from the ALU back to registers. The PC and IR are also shown with data flow arrows.

Engenharia de Sistemas Embarcados 15
 Grupo de Engenharia da Computação - CIn / UFPE

Sub-operações da unidade de controle

- **Decodificação**
 - Determina o significado de cada instrução

The diagram shows the processor control unit and datapath. The control unit contains a controller, PC, and IR. The datapath contains registers, ALU, and I/O. Memory is connected to the datapath. Arrows indicate the flow of data: from memory to registers, from registers to the ALU, and from the ALU back to registers. The PC and IR are also shown with data flow arrows.

Engenharia de Sistemas Embarcados 16
 Grupo de Engenharia da Computação - CIn / UFPE

Sub-operações da unidade de controle

- **Busca de operandos**
 - Transfere dados da memória para o registrador do datapath

The diagram shows the processor control unit and datapath. The control unit contains a controller, PC, and IR. The datapath contains registers, ALU, and I/O. Memory is connected to the datapath. Arrows indicate the flow of data: from memory to registers, from registers to the ALU, and from the ALU back to registers. The PC and IR are also shown with data flow arrows.

Engenharia de Sistemas Embarcados 17
 Grupo de Engenharia da Computação - CIn / UFPE

Sub-operações da unidade de controle

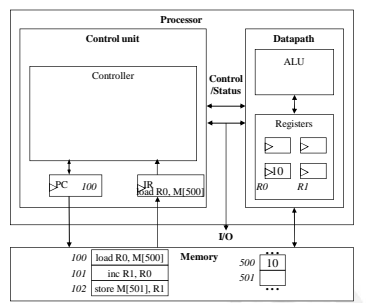
- **Execução**
 - ULA executa a operação requerida
 - A instrução load mostrada neste exemplo não requer uma operação lógica aritmética

The diagram shows the processor control unit and datapath. The control unit contains a controller, PC, and IR. The datapath contains registers, ALU, and I/O. Memory is connected to the datapath. Arrows indicate the flow of data: from memory to registers, from registers to the ALU, and from the ALU back to registers. The PC and IR are also shown with data flow arrows.

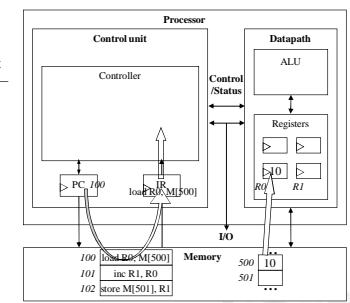
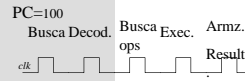
Engenharia de Sistemas Embarcados 18
 Grupo de Engenharia da Computação - CIn / UFPE

Sub-operações da unidade de controle

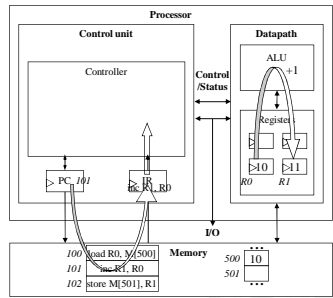
- Armazenagem de resultados
 - Escrita de resultados armazenados em registradores
 - A instrução load mostrada neste exemplo não requer uma operação lógica aritmética



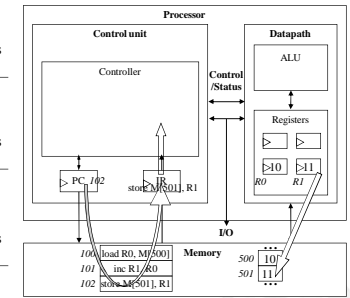
Ciclos de Instrução



Ciclos de Instrução

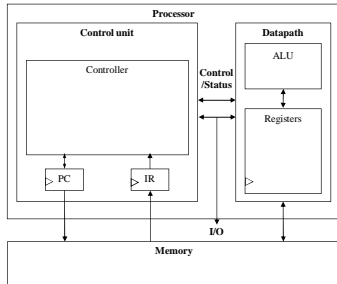


Ciclos de Instrução



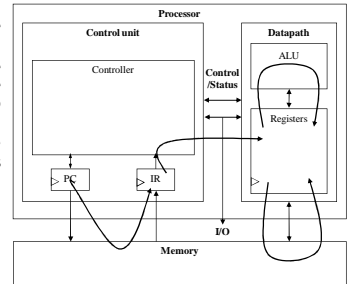
Considerações de Arquitetura

- **Processador N-bit**
 - N-bit ALU, registradores, barramento, interface de dados da memória
 - Embarcado: comuns 8-bit, 16-bit, 32-bit
 - Desktop/servidores: 32-bit, 64-bit
- PC tamanho determina o espaço de endereçamento



Considerações de Arquitetura

- **Frequência de Clock**
 - Inverso do período de clock
 - Deve ser mais longo que o atraso máximo entre registradores de todo o processador
 - Acesso de memória é com frequência o mais longo



Pipelining: Aumentando o desempenho

Classe de Instrução	Memória de Instruções	Leitura de Registrador	Operação na ULA	Memória de Dados	Escrita no REG	Total
Formato R (add, sub, and, or, slt)	2	1	2	0	1	6 ns
Load word (lw)	2	1	2	2	1	8 ns
Store word (sw)	2	1	2	2	0	7 ns
Branch (beq)	2	1	2	0	0	5 ns
Jump (jmpl)	2	0	0	0	0	2 ns

- Comparando
 - 24ns (sem pipeline) x 14ns (com pipeline)

Engenharia de Sistemas Embarcados 25

Pipelining: Aumentando o Throughput de Instrução

- Como melhorar o desempenho de máquinas implementadas como pipeline?
 - Aumentar o número de estágios do pipeline
 - Superpipeline
 - Replicar recursos para executar instruções em paralelo
 - Superescalar

Engenharia de Sistemas Embarcados 26

Pipelining: Aumentando o desempenho

- Superpipeline
 - Limitações no tamanho do pipeline
 - Hazards de dados: pipeline maior => mais paradas
 - Hazards de controle: pipeline maior => saltos mais lentos
 - Tempo dos registradores do pipeline: Limita o tempo mínimo por estágio (clock)

Engenharia de Sistemas Embarcados 27

Pipelining: Aumentando o desempenho

- Superescalar
 - Execução simultânea das instruções
 - Aritméticas, Loads e Stores
 - Aplicável a máquinas RISC e CISC
 - RISC: melhor uso efetivo
 - CISC: implementação mais difícil

Engenharia de Sistemas Embarcados 28