

A Arquitetura do Computador Visível

O Computador Visível é um computador de uso geral, com 512 bytes de memória, palavra de 16 bits, conjunto de instruções formado por 17 instruções básicas, 3 modos de endereçamento (imediato, direto e indireto), dois modos de acesso à memória (absoluto ou relativo), com velocidade de clock variando de 0,25 Hz até 16 Hz.

A velocidade de funcionamento do computador está basicamente limitada pela capacidade de processamento da CPU na qual o applet está executando.

Os detalhes arquiteturais da CPU do Computador Visível podem ser vistos na janela **Central Processing Unit**. Nesta janela são mostrados detalhes praticamente completo da CPU, inclusive da unidade de controle controlada por micro código.

As instruções do VC têm tamanho fixo de 16 bits, onde os 7 bits de alta ordem contêm o operador (instrução + modo de endereçamento + modo de acesso à memória). Os 9 bits de baixa ordem contem o operando, normalmente um endereço de memória.

O esquema mostrado na janela da CPU pode ser dividido em 3 partes:

Linhas do Barramento Principal

O barramento principal do VC fica localizado à esquerda da janela, e é formado por 7 setas (linhas do barramento) que descem de cima a baixo na janela.

- reset

A linha de **reset** é utilizada para provocar um reset geral na CPU. Ela está conectada ao botão **Start/Reset**, localizado na janela do applet. Do barramento reset sai uma linha que se conecta à unidade chamada de **Nano Sequencer**, que é de onde se origina todo o controle da CPU.

- ready

É utilizada para que os dispositivos de entrada e saída de dados informem à CPU o término de uma operação de E/S (Entrada/Saída). O estado dos dispositivos de E/S estão representado na janela **Input/Output**. Do barramento **ready** sai uma linha que se conecta à unidade chamada de **mInstr Sequencer (Micro Instruction Sequencer)**. O Micro Instruction Sequencer é responsável pelo sequenciamento das microinstruções que controlam o fluxo de dados dentro da CPU.

- wait

É utilizada para que a CPU indique aos dispositivos de entrada que ela está aguardando a finalização de uma operação de E/S. O sinal de wait é originado do **MicroInstructionRegister**, ou Registrador de Micro Instruções.

- read

É utilizada para que a CPU indique à memória que ela deve ler a posição indicada pelo conteúdo do **Memory Addr Reg** e colocar o valor lido no **Memory Data Reg**. O sinal read é originado do **MicroInstructionRegister**.

- write

É utilizada para que a CPU indique à memória que ela deve gravar na posição indicada pelo conteúdo do **Memory Addr Reg** o valor contido no **Memory Data Reg**. O sinal write é originado do **MicroInstructionRegister**.

- data (Bus)

Barramento de 16 bits É utilizado para conectar CPU e Memória, permitindo a transferência de dados entre estas unidades.

- address (Bus)

Barramento de 9 bits ($2^9 = 512 =$ tamanho da memória), Utilizado para conectar CPU e Memória, permitindo a transferência de endereços entre estas unidades.

Registadores, Unidade Aritmética e Lógica, Barramento Interno da CPU

A arquitetura da CPU é composta por 9 unidades funcionais: sete registradores, uma unidade aritmética e lógica, e o barramento interno da CPU. Ao barramento interno da CPU se conectam várias linhas de entrada e saída de dados, que interconectam todas as outras unidades.

A função de cada unidade é mostrada abaixo:

- Memory Addr Reg

Armazena endereços de memória, que são utilizados pela unidade de memória, para realizar as operações de leitura e escrita de palavras. Apenas os 9 bits menos significativos deste registrador são transferidos para o **Data Bus**, através da linha **to address bus**.

- Memory Data reg

Armazena valores que são lidos da memória (read), ou escritos na memória (write).

- Accumulator

Do ponto de vista do programador de baixo nível (assembler), que trabalha com o conjunto de instruções da CPU, este é o principal registrador da CPU. É a ele que fazem referências praticamente todas as instruções d CPU.

- A.L.U (Arithmetic and Logic Unit, ou Unidade Aritmética e lógica)

Recebe dados provenientes do registrador **Accumulator** e do **CPU Bus (barramento interno da CPU)** e realiza operações conforme o sinal **alu control**, proveniente do **Micro Instruction register**. O valor resultante da operação da ALU é armazenado em um registrador interno da ALU, cujo valor é mostrado no quadro visível da ALU.

- Auxiliary register

Registrador auxiliar, não acessível ao programador assembler, que armazena valores intermediários necessários ao funcionamento interno da CPU.

- Segment Register

Armazena índices de posicionamento na memória, utilizados pelas instruções que fazem referência relativa à memória. É útil para o escalonamento de código relocável.

- Instruction Pointer

Contém o endereço da locação de memória de uma instrução. É com base neste endereço que as instruções são carregadas da memória para o Registrador de Instruções.

- **Instruction Register**

Contém a instrução que será, ou já foi, interpretada pela CPU. Os sete bits de mais alta ordem deste registrador (opcode) são transferidos através da linha **Opcode** para o **Instruction Decoder**, este último localizado na unidade de controle.

- **CPU Bus (barramento interno da CPU)**

O **CPU Bus** permite o fluxo de dados entre todas as unidades funcionais da CPU. A fluxo de dados é controlado pelas diversas linhas que saem de várias posições no

MicroInstruction Register. Quando os dados fluem de uma unidade para outra a linha que conecta estas duas unidades se acende e apaga três vezes consecutiva, e o valor é então transferido de uma unidade para outra.

Um mnemônico que indica que tipo de fluxo de dados está sendo, ou será efetuado na CPU é mostrado na parte interna do Micro Instruction Register.

Control Unit (Unidade de Controle)

A unidade de controle é composta por 11 unidades funcionais, cujo objetivo final é colocar no MicroInstructionRegister uma micro instrução que provoca o fluxo de dados adequado entre os registradores e a ALU, bem como a integração entre os sinais de controle provenientes do barramento principal do VC.

Os elementos desta unidade são:

- **MicroInstructionRegister**

É um registrador de 41 bits, que pode ser dividido em x seções, de acordo com o intervalo dos bits (os bits de menor ordem estão localizados à direita, começando com o valor 0):

- **END MICROCODE (bits 40-40)**
Este bit indica que a micro instrução corrente é a última a ser executada para interpretação de uma instrução.
- **ALU CONDITIONS (bits 39-36)**
Indicam quais os estados que são pré-codificação para a execução desta micro instrução.
- **JUMP ADDRESS (bits 35-26)**
Se o valor destes bits for diferente de 0, então a próxima micro instrução a ser executada será a indicada por este endereço.
- **WAIT (bits 25-25)**
A CPU está aguardando final de operação de E/S.
- **READ (bits 24-24)**
A CPU solicita à unidade de memória a leitura da palavra cujo endereço está disponível no Memory Address Register.
- **WRITE (bits 23-23)**

A CPU solicita à unidade de memória a escritura da palavra cujo endereço está disponível no Memory Address Register, com o conteúdo disponível no Memory Data Register.

- ALU CONTROL (bits 22-15)
Indicam que a ALU deve realizar uma dentre as 8 possíveis funções que a ALU disponibiliza. De acordo com o valor não nulo de um dos bits, as seguintes operações serão realizadas:
 - COMPARE (bit 22)
Compara o valor do acumulador com o valor do CPU Bus e ajusta os flags da ALU de acordo com o resultado: EQ ($ACC == CPU\ Bus$), GEQ ($ACC \geq CPU\ Bus$), LEQ ($ACC \leq CPU\ Bus$), ZERO ($ACC == 0$).
 - ADD (bit 21)
Soma Acc com CPU Bus e coloca o resultado no registrador da ALU.
 - SUB (bit 20)
Subtrai de Acc o valor do CPU Bus e coloca o resultado no registrador da ALU.
 - XOR (bit 19)
Coloca no registrador da ALU o resultado do XOR binário entre Acc e CPU Bus.
 - AND (bit 18)
Coloca no registrador da ALU o resultado do AND binário entre Acc e CPU Bus.
 - OR (bit 17)
Coloca no registrador da ALU o resultado do OR binário entre Acc e CPU Bus.
 - NEG (bit 16)
Coloca no registrador da ALU o resultado da inversão binária do valor contido em Acc.
 - INC (bit 15)
Coloca no registrador da ALU o resultado da soma do valor contido em Acc com 2.
- DATA FLOW BITS (bits 14-0)
Estabelecem a ligação entre os componentes da arquitetura da CPU e o CPU Bus. O significado de cada um dos bits é:
 - bit 14

ALU input \leq CPU Bus

- bit 13
CPU Bus \leq ALU output
- bit 12
CPU Bus \leq Accumulator
- bit 11
Accumulator \leq CPU Bus
- bit 10
CPU Bus \leq Memory Data Register
- bit 9
Memory Data Register \leq CPU Bus
- bit 8
Memory Address Register \leq CPU Bus
- bit 7
CPU Bus \leq bits 9 a 0 do Instruction Register
- bit 6
Instruction Register \leq CPU Bus
- bit 5
CPU Bus \leq Instruction Pointer
- bit 4
Instruction Pointer \leq CPU Bus
- bit 3
CPU Bus \leq Segment Register
- bit 2
Segment Register \leq CPU Bus
- bit 1
CPU Bus \leq Auxiliary Register
- bit 0
Auxiliary Register \leq CPU Bus

- Instr Decoder (Instruction Decoder, ou Decodificador de Instruções)

Ao receber um sinal vindo da Nano Control Unit (unidade localizada no canto inferior esquerdo da unidade de controle, que contém o Nano Sequencer e o Nano Decoder), esta unidade disponibiliza para o barramento da unidade de controle o endereço inicial da micro

instrução que corresponde à execução da instrução cujo **Opcode** está disponibilizado no Instruction Register.

- **MicroInstr Decoder** (Micro Instruction Decoder), ou decodificador de micro instruções) Ao receber um sinal vindo da Nano Control Unit, esta unidade disponibiliza para o **mInstrSequencer** uma microinstrução, cujo endereço foi determinado pelo valor contido no registrador **m Instr Pointer**.

- **m Instr Pointer** (Micro Instruction Pointer)

Contém o endereço da próxima microinstrução que deve ser disponibilizada pelo **MicroInstr Decoder**. O **mInstr Sequencer** eventualmente também faz uso deste valor.

- **Fetch Instr** (register)

É um registro apenas de leitura, e contém o endereço inicial da micro instrução que realiza a busca de uma instrução na memória.

- **Incr I.P.** (register)

É um registro apenas de leitura, e contém o endereço inicial da micro instrução que incrementa o valor do **Instruction Pointer**

- **Reset CPU** (register)

É um registro apenas de leitura, e contém o endereço inicial da micro instrução que realiza o Reset/Start da CPU **Instruction Pointer**

- **Nano Control Unit**

É o ponto chave de controle da CPU. Controla todo o fluxo de dados entre os elementos da unidade de controle. Esta unidade trabalha auxiliada pelo **mInstrSequencer**, e é composta por duas subunidades: o **Nano Sequencer** e o **Nano Decoder**.

O NanoSequencer realiza a temporização básica dos ciclos de execução de uma instrução. Após receber o sinal de reset e controlar o reset da CPU, esta unidade fica ciclicamente executando entre os estados: FETCHING, IP=IP+2 e EXEC INST. Os estados que o microsequencer pode assumir são:

- **RESET ?** - Neste estado o NanoSequencer está aguardando o sinal de reset.
- **RESETTING** - Neste estado o NanoSequencer está controlando o reset da CPU. O microprograma apontado pelo registrador **Reset CPU** é carregado.
- **FETCHING** - Neste estado o NanoSequencer está controlando a busca da instrução em memória. O micro programa apontado pelo registrador **Fetch Instr** é carregado.
- **IP=IP+2** - Neste estado o NanoSequencer está controlando o incremento do ponteiro de instruções, preparando-se para buscar esta instrução no próximo ciclo. O micro programa apontado pelo registrador **Icr I.P.** é carregado.
- **EXEC INST** - Neste estado o NanoSequencer está controlando a execução da instrução propriamente dita. Neste caso o endereço da micro instrução a ser executada é determinado pelo **Instr Decoder**.

O Nano Decoder carrega, a pedido do NanoSequencer, um conjunto de valores no **Nano**

Instruction Register (canto superior direito da Nano Control Unit). Conforme os valores dos bits deste registrador são ativados controles e fluxos de dados entre os elementos da unidade de controle.

- **mInstr Sequencer** (Micro Instruction Sequencer, ou seqüenciador de micro instruções) Esta unidade é responsável basicamente por receber uma micro instrução, proveniente do **Micro Instruction Decoder** e decidir se deve ou não executa-la (carregá-la no MicroInstruction register), conforme o estado da alu (linha **alu state**).

O Micro Instruction Sequencer é ativado pelo sinal **exec** proveniente da nano Control Unit (segunda linha à esquerda, que sai do **Nano Decoder**). Após a carga da micro instrução no MicroInstructionRegister, o Micro Instruction Sequencer sinaliza para o **Nano Sequencer** o final da micro instrução (sinal **end**), ou a solicitação de execução da próxima micro instrução (sinal **next**).

O Micro Instruction Sequencer é ainda responsável por bloquear o funcionamento normal da CPU, ficando em estado de WAIT, aguardando que os dispositivos de E/S sinalizem o final da operação de E/S. O final de uma operação de E/S é indicado através do sinal **ready**.

Autor: Yuri Bueno Fontenla
Contato: gold_fox1@hotmail.com