

Instituto Federal Sul-Rio-Grandense – Campus Venâncio Aires

Curso Superior em Análise e Desenvolvimento de Sistemas

Arquitetura e Organização de Computadores

Professor Fernando Herrmann

Zilog Z80

Diogo Henrique Schlosser, Guilherme Johann Reckziegel Nunes e Mateus Roberto Algayer

História

No ano de 1969 a Intel foi contratada pela companhia japonesa Busicom para produzir uma calculadora para desktop. A Intel sugeriu que a calculadora precisaria ser feita em volta de um mecanismo de computação generalizada de chip-único, e assim nasceu o primeiro microprocessador, o 4004, que utilizava até 4 Bits.

Intel criou uma variação melhor do chip, o 4040, que foi criado a partir do 4004 mas com um design mais avançado. Depois disso teve o 8008, o primeiro chip usando 8 Bits, depois o 8080, uma versão melhorada do 8008, que foi utilizada no primeiro computador doméstico, o Altair 8800.

Em 1975 um dos funcionários da Intel que trabalhou com o chip 4004 e seus predecessores deixou a Intel e se uniu a Masatoshi Shima para criar a empresa Zilog. Eles então criaram um microprocessador compatível ao 8080, ele consegue executar todas as 78 instruções do 8080 da exata mesma forma que o chip da Intel, mas ainda conseguia executar mais 120 execuções extras, mais registradores e possuía uma conexão mais simplificada com o hardware. Esse foi o nascimento do Z80

O Z80 original saiu em Julho de 1976, depois disso saiu várias versões utilizando a mesma estrutura, mas com maior velocidade:

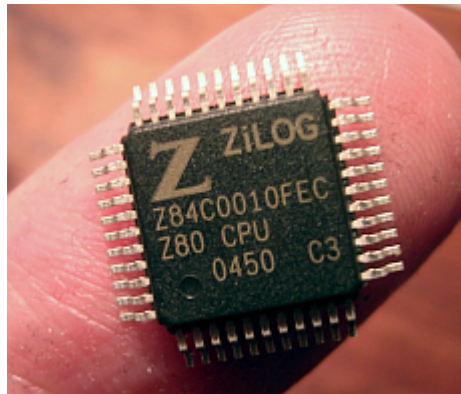
- Z80: executava com um clock rate de 2.MHz;
- Z80A: executava com um clock rate de 4MHz;
- Z80B: executava com um clock rate de 6MHz;
- Z80H: executava com um clock rate de 8MHz.

Durante os anos 70 e 80 muitas empresas produziram máquinas baseadas no chip da Zilog, pois ele conseguia executar o 8080 sem a necessidade de alterar os códigos do SO CP/M. Também foi usado em vários videogames da época.

Periféricos que usavam o Z80:

- Radio Shack TRS-80;
- Sinclair ZX-80, ZX-81 e ZX-Spectrum;
- Amstrad CPC;
- Commodore 128;

- MSX;
- Naja Micro;
- Texas Instruments modelos TI-73, TI-81, TI-82, TI-83, TI-84, TI-85 e TI-86;
- Players de MP3, de marca genérica;
- Várias placas de arcade, como a CPS-1 e CPS-2;
- Sega Master System;
- Sega Mega Drive;
- Nintendo Game Boy.



Arquitetura

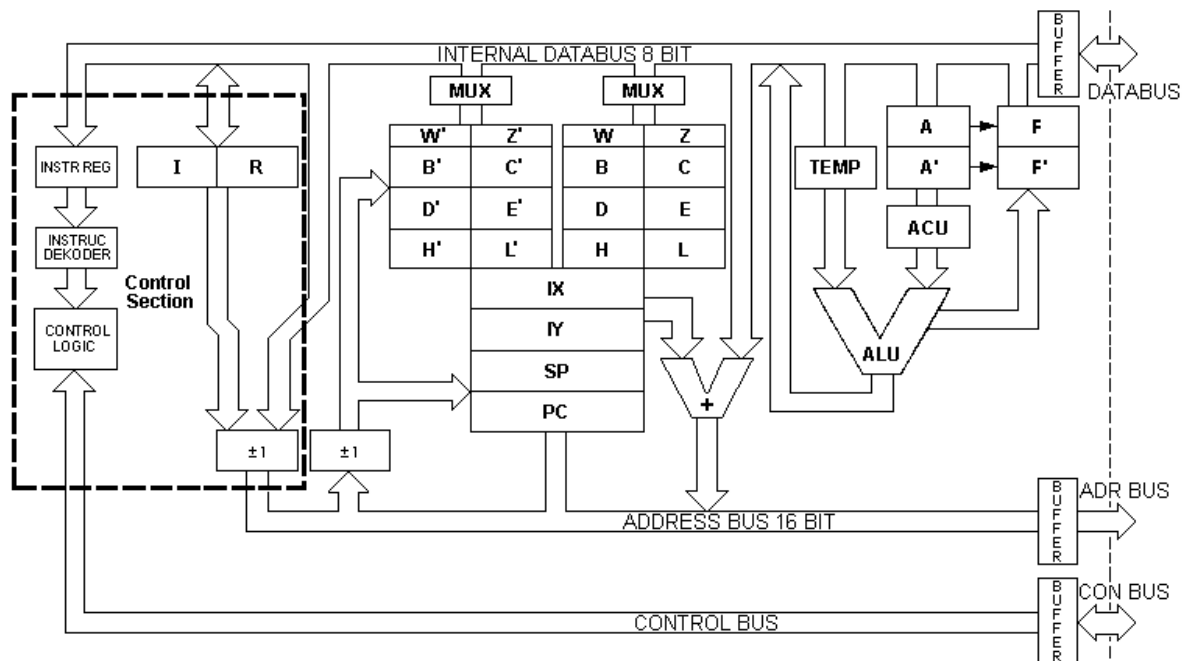


Figura 1 - Arquitetura do Zilog Z80

Fonte: <http://marmsx.msxall.com/cursos/assembly/assembly4.html>

O Zilog Z80 é um processador com arquitetura de 8 bits e clock de 3,58 MHz sendo considerado um processador de arquitetura CISC. Como vários processadores da época, o Zilog Z80 possuía uma tecnologia chamada de *fetch/execute overlapping*, tecnologia pela qual o processador era capaz de buscar a próxima instrução na memória(fetch) enquanto executava a instrução atual.

Para armazenar informações e realizar as diversas operações que este processador permite realizar ele utiliza de 20 registradores(12 registradores de primeiro plano e 8 registradores de segundo plano) de 8 bits e 4 registradores de 16 bits, cada registrador pode armazenar até 8 bits de memória, podendo utilizar até 16 bits de memória se utilizados em pares.

Registradores		
A	Registrador usado pela ULA para armazenar resultados aritméticos e para comunicação externa.	8 bits
B	Registrador considerado como comum, mas que pode ser utilizado para contagem de laços para determinadas instruções como a DJNZ.	8 bits
C	Registrador considerado comum, pode ser par do Registrador B	8 bits

	formando um registrador de 16 bits.	
D	Registrador comum.	8 bits
E	Registrador considerado comum, pode ser par do Registrador D formando um registrador de 16 bits.	8 bits
W	Registrador usado internamente pelo processador para certas operações	8 bits
Z	Registrador usado internamente pelo processador para certas operações	8 bits
HL	Registrador de 16 bits (são em realidade dois registradores de 8 bits, mas são considerados como 1 registrador de 16 bits).	8 bits
F	Registrador de flags. Armazena propriedades resultantes de cálculos aritméticos	8 bits
IX	Registrador de índice, permite deslocamentos fixos para variáveis. Ex: $IX + 1$	16 bits
IY	Registrador de índice, permite deslocamentos fixos para variáveis. Ex: $IY + 1$	16 bits
PC	Registrador que guarda um ponteiro para a memória principal, usada para determinar qual o próximo passo a ser executado.	16 bits
SP	Registrador que guarda um ponteiro para a memória principal contendo o endereço do topo da pilha de execução.	16 bits

Registradores alternativos(linha ou segundo plano):

A', B', C', D', E', F', H', L', W', Z'

São registradores adicionais de 8 bits que não podem ser utilizados normalmente, apenas através dos comandos EX e EXX que alteram o valor dos registradores linhas com seus respectivos pares(AF, BC, DE, HL). Ex: A'F' é alterada com AF e vice-versa.

O Zilog Z80 como vários outros processadores possui um registrador para guardar flags sobre algumas propriedades encontradas pelo processador ao realizar determinadas operações lógicas e aritméticas, abaixo uma tabela com as flags armazenadas por ele.

Flags			
Bit	Flag		
7	S	Sign	Flag que indica se o resultado da operação é negativo(1) ou positivo(0)
6	Z	Zero	Flag que indica se o resultado da operação é zero(1) ou não-zero(0)
5	-	Não possui	
4	H	Half-Carry	Indica com 1 quando a operação estourou no bit 3
3	-	Não possui	
2	P	Parity/Overflow	Indica se o número de bits ligados do resultado é par(1) ou ímpar(0)
1	N	Subtraction	Indica se a última operação realizada foi subtração(1) ou adição(0)
0	C	Carry	Indica com 1 quando a operação estourou no bit 7

Conjunto de instruções

Possui cerca de 256 instruções, dentre suas principais funções estão:

- Load and exchange
- Block transfer and search
- Arithmetic and logical
- Rotate and shift
- Bit manipulation (set, reset, test)
- Jump, call and return
- Input/output
- Basic CPU control

Exemplos de instruções¹:

LD *r*, (HL)

O conteúdo da posição de memória (HL) é carregada para o registrador *r*. O símbolo *r* identifica os registradores A, B, C, D, E, H ou L.

LD (HL), *r*

O conteúdo do registrador *r* é carregado para a posição de memória especificada pelo conteúdo do par de registradores HL. O símbolo *r* identifica os registradores A, B, C, D, E, H ou L.

PUSH *qq*

O conteúdo do par de registradores *qq* é empurrado para a pilha de memória externa último a entrar, primeiro a sair (LIFO). O registrador do ponteiro da pilha (SP) guarda o endereço de 16 bits do topo atual da pilha.

Esta instrução é usada para salvar dados dos registradores temporariamente.

POP *qq*

Os dois bytes do topo da pilha de memória externo último a entrar, primeiro a sair (LIFO) são carregados para o par de registradores *qq*. O registrador do ponteiro da pilha (SP) guarda o endereço de 16 bits do topo atual da pilha.

Esta instrução é usada para carregar dados salvos temporariamente.

EX AF, AF'

Os conteúdos dos pares de registradores AF e AF' são trocados entre si, de modo que AF recebe o valor de AF' e AF' o valor de AF.

EXX

Troca o conteúdo de todos os pares de registradores por suas equivalentes linhas. De modo que BC, DE e HL são trocados por BC', DE' e HL' e BC', DE' e HL' são trocados para BC, DE e HL.

¹ A documentação das instruções do Zilog Z80 podem ser encontradas [aqui](#).

CALL nn, CALL cc, nn

CALL nn chama uma subrotina para o topo da stack, já o CALL cc, nn testa se a flag cc está ligada, se sim chama a subrotina para o topo da stack, caso não esteja segue a stack de forma regular.

Demais instruções do Zilog podem ser encontradas na documentação oficial do Zilog neste [link](#).

Emulador/Simulador

Para simular e montar o código utilizamos o Z80_workbench.

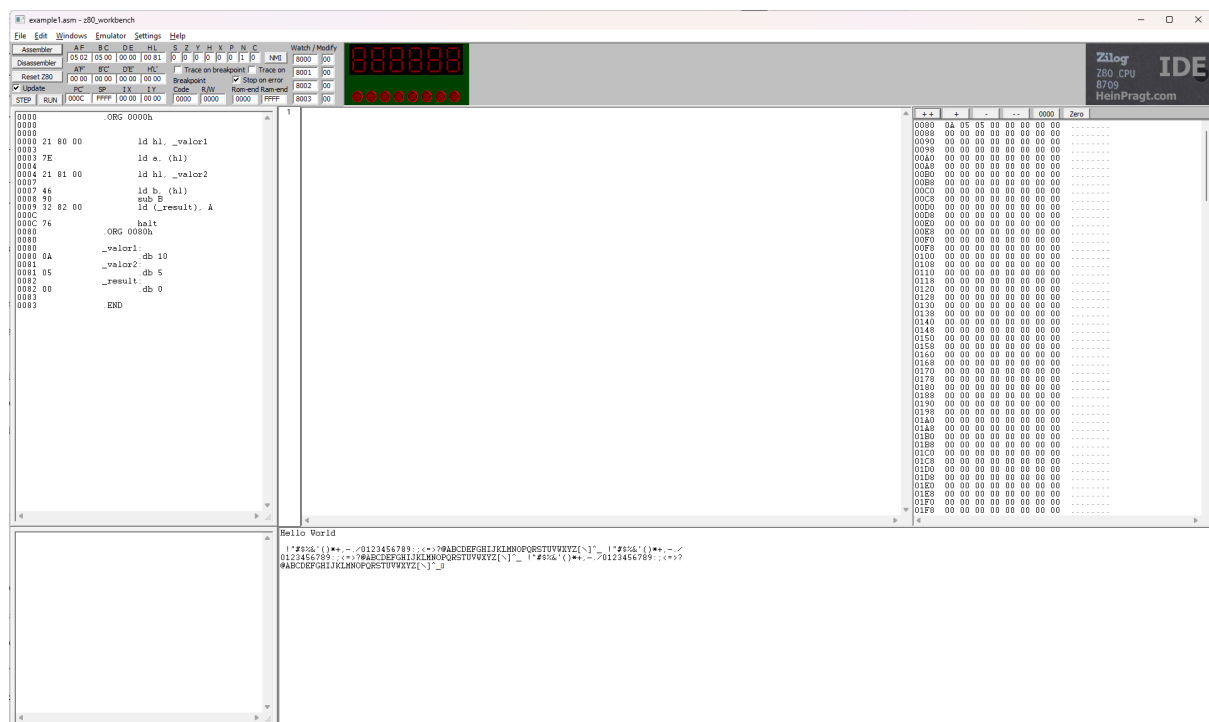


Figura 2 - Tela do Z80_workbench

Fonte: Os autores

Faça o download [aqui](#).

Com o Z80_workbench aberto, pode-se programar² usando o box do meio, após terminar o código deve-se apertar o botão Assembler no canto superior esquerdo, caso haja erros de compilação os mesmos serão mostrados no box na esquerda inferior, com tudo correto basta apertar o botão RUN(recomenda-se desmarcar a opção “update” e marcar a opção “trace on”).

Após apertar o botão RUN é possível ver o que está nos registradores na parte superior e o que está na memória no box da direita, pode-se ao invés do RUN utilizar o botão STEP, onde será possível ver o passo-a-passo da execução.

² A documentação das instruções do Zilog Z80 podem ser encontradas [aqui](#).

Programas desenvolvidos em Assembly

Programa na arquitetura do Z80 para fazer a operação $10 - 5$

```
.ORG 0000h ; Seta a posição onde o código de execução será gravado
ld hl, _valor1 ; Carrega o endereço de _valor 1 em hl
ld a, (hl) ; Lê o endereço para onde hl aponta e guarda em A
ld hl, _valor2 ; Carrega o endereço de _valor2 em hl
ld b, (hl) ; Lê o endereço para onde hl aponta e guarda em B
sub B ; Executa a subtração
ld (_result), A ; Armazena o valor em A no ponteiro _result
halt ; parar a execução

.ORG 0080h ; Seta a posição onde as variáveis serão gravadas
_valor1: ; Define _valor1 como 10 na posição 0x0080
.db 10
_valor2: ; Define _valor2 como 5 na posição 0x0081
.db 5
_result: ; Define _result como 0 na posição 0x0082
.db 0
.END ;; Fim do programa
;; 0x0080 = 0A16(1010), 0x0081 = 0516(510), 0x0082 = 0016(010)
```

Programa na arquitetura do Z80 para fazer a operação $10 + 5$

```
.ORG 0000h ; Seta a posição onde o código de execução será gravado
ld hl, _valor1 ; Carrega o endereço de _valor 1 em hl
ld a, (hl) ; Lê o endereço para onde hl aponta e guarda em A
ld hl, _valor2 ; Carrega o endereço de _valor2 em hl
ld b, (hl) ; Lê o endereço para onde hl aponta e guarda em B
add A, B ; Executa a adição
ld (_result), A ; Armazena o valor em A no ponteiro _result
halt ; parar a execução

.ORG 0080h ; Seta a posição onde as variáveis serão gravadas
_valor1: ; Define _valor1 como 10 na posição 0x0080
.db 10
_valor2: ; Define _valor2 como 5 na posição 0x0081
.db 5
_result: ; Define _result como 0 na posição 0x0082
.db 0
.END ;; Fim do programa
;; 0x0080 = 0A16(1010), 0x0081 = 0516(510), 0x0082 = 0016(010)
```

Referências

SHVETS, Gennadiy. Zilog Z80 microprocessor family. [S. l.], 8 nov. 2022. Disponível em: <https://www.cpu-world.com/CPUs/Z80/>. Acesso em: 27 jan. 2023.

THE Z80 Microprocessor. [S. l.], 1998. Disponível em: <https://landley.net/history/mirror/cpm/z80.html>. Acesso em: 22 jan. 2023.

CURSO de Assembly: Conhecendo o Processador Z80. [S. l.], 2020. Disponível em: <http://marmsx.msxall.com/cursos/assembly/assembly4.html>. Acesso em: 22 jan. 2023.

SCHERRER, Thomas. Hardware - Software - Utilities - FAQ - Docs for Z80-Family. [S. l.], 2023. Disponível em: <http://www.z80.info/z80arki.htm>. Acesso em: 22 jan. 2023.

Z80 Microprocessors Z80 CPU User Manual. [S. l.: s. n.], 2016. Disponível em: <https://www.zilog.com/docs/z80/um0080.pdf>. Acesso em: 22 jan. 2023.