Simulador AtMega 328-p

Raul Ramires[1]

[1]Departamento de Informática – Universidade Estadual de Maringá (UEM)
Maringá – PR – Brazil

ra82293@uem.br

Resumo. *O microcontrolador AtMega 328-p é amplamente usado em sistemas embarcados, como por exemplo o Arduino Uno. Esse microcontrolador permite realizar a leitura de vários tipos de sensores, controlar motores e outros dispositivos eletrônicos. A implementação de um ambiente de simulação permite ao usuário projetar circuitos e programações no microcontrolador sem a necessidade de possuir a unidade física.*

1. Introdução

A implementação de um ambiente de simulação para o microcontrolador AtMega 328-p consiste em primeiramente implementar a execução do conjunto de instruções e as estruturas de registradores e memórias. As instruções do microcontrolador podem ser encontradas no manual [Atmel 2016].

2. Arquitetura Atmega 328-p

O microcontrolador Atmega 328-p possui 3 memórias diferentes, sendo elas:

- 32Kb de memória *flash*;
- 2Kb de memória SRAM;
- 1Kb de memória EEPROM.

Esse microcontrolador ainda possui 32 registradores de propósito geral de 8 bits, sendo que alguns trabalham em pares para permitir endereçamento de 16 bits. Registradores X, Y e Z atuam como ponteiros de 16 bits para o endereçamento do espaço de dados, o que permite um cálculo de endereços eficiente.

A Figura 1 mostra o diagrama de blocos do microcontrolador.

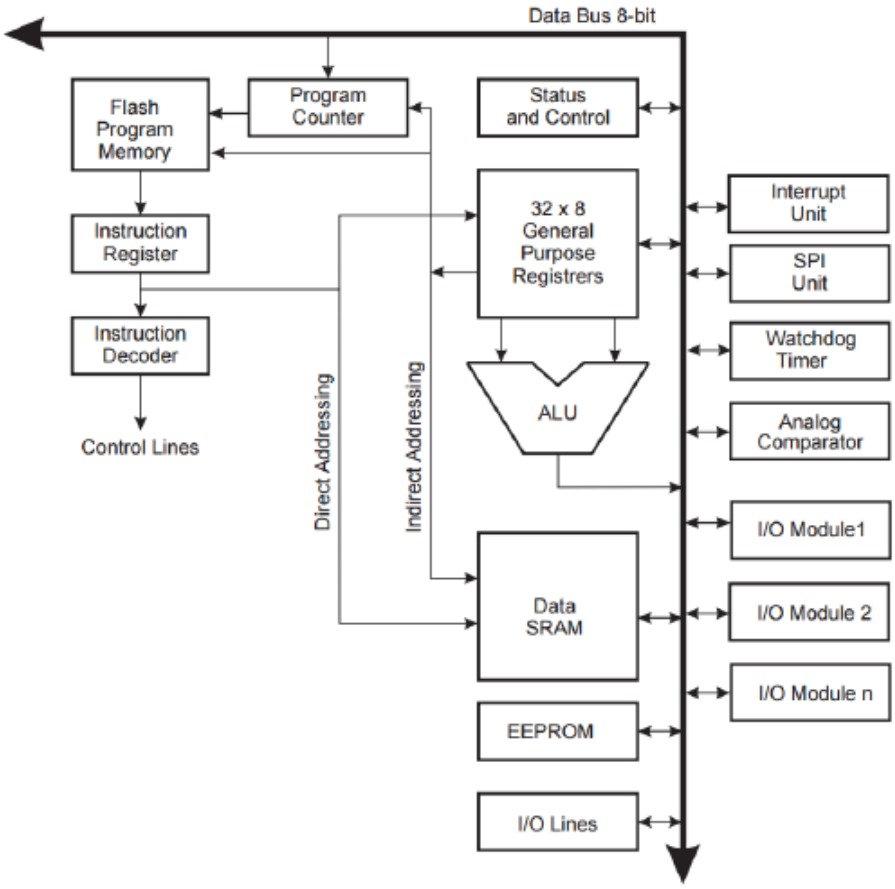

Figura 1. Diagrama de blocos do microcontrolador.

O registrador de estado (SREG) possui 8 bits e armazena algumas *flags*, de acordo com a Figura 2.

Bit	7	6	5	4	3	2	1	0	
0x3F (0x5F)	I	T	H	S	V	N	Z	C	SREG
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Figura 2. Registrador de Estado SREG.

As *flags* do registrador SREG são alteradas na execução de uma instrução, sendo que cada instrução altera *flags* específicas de acordo com o manual. As *flags* são as seguintes:

I $\rightarrow$ Interrupção global;
T $\rightarrow$ Armazenamento de bit de cópia;
H $\rightarrow$ *Half Carry*;
S $\rightarrow$ Bit de sinal, $S = N \oplus V$;
V $\rightarrow$ *Overflow* de complemento de 2;
N $\rightarrow$ Negativo;

Z $\rightarrow$ Zero;

C $\rightarrow$ *Carry*.

O microcontrolador possui um registrador PC (*Program Counter*) de 14 bits que tem o propósito de apontar para a instrução que será executada. O valor de PC é incrementado a cada instrução executada, podendo ainda ser alterado por instruções de saltos.

3. Implementação

A implementação da execução do conjunto de instruções foi feita na linguagem C. Os registradores de propósito geral foram definidos como um *array* de 32 posições do tipo *uint_8*. O registrador SREG foi implementado como uma *struct* com os campos das *flags* com o tipo de dado *uint_8*.

A Figura 3 mostra a definição desses registradores no arquivo *registers.h*.

```
#include <stdint.h>

uint8_t R[32];

struct SREG{
    uint8_t I,T,H,S,V,N,Z,C;
}SREG;

uint16_t PC;
```

Figura 3. Definições dos registradores.

Ainda não foram implementadas todas as instruções disponíveis no microcontrolador, mas estão implementadas uma grande parte das instruções. As instruções estão programadas nos arquivos *instruction_set.c* e *instruction_set.h*.

3.1. Instruções Aritméticas

Foram implementadas as instruções aritméticas de soma com e sem *carry*. A Figura 4 mostra o código da instrução de soma sem *carry*.

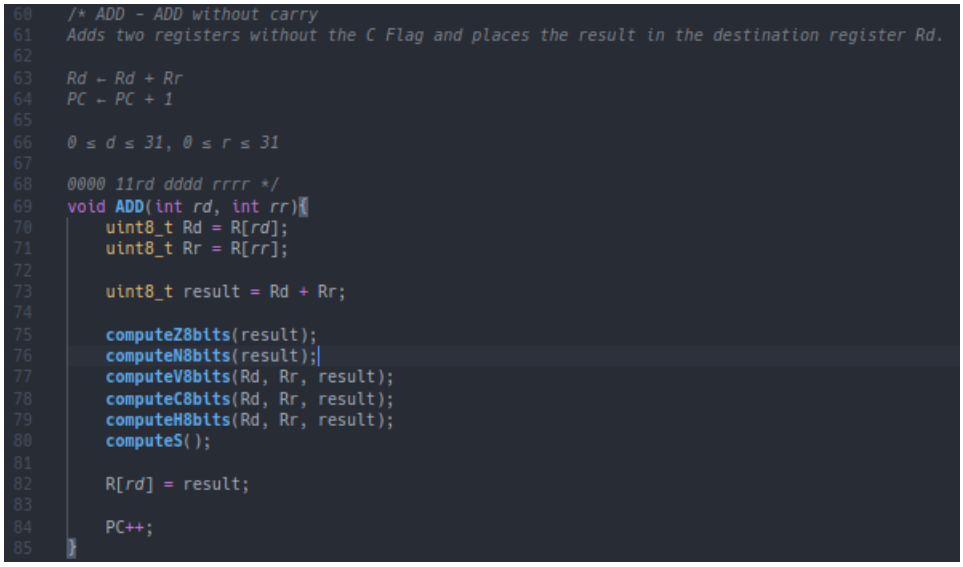

Figura 4. Instrução de soma sem *carry*.

As funções auxiliares para cálculo das *flags* do registrador de estado SREG estão implementadas nos arquivos *functions.c* e *functions.h*.

3.2. Instruções Lógicas

Foram implementadas a maior parte das instruções lógicas, como por exemplo a operação *AND* entre dois registradores, alterando as devidas *flags* do registrador de estado. A Figura 5 mostra o código da operação *AND* entre dois registradores.

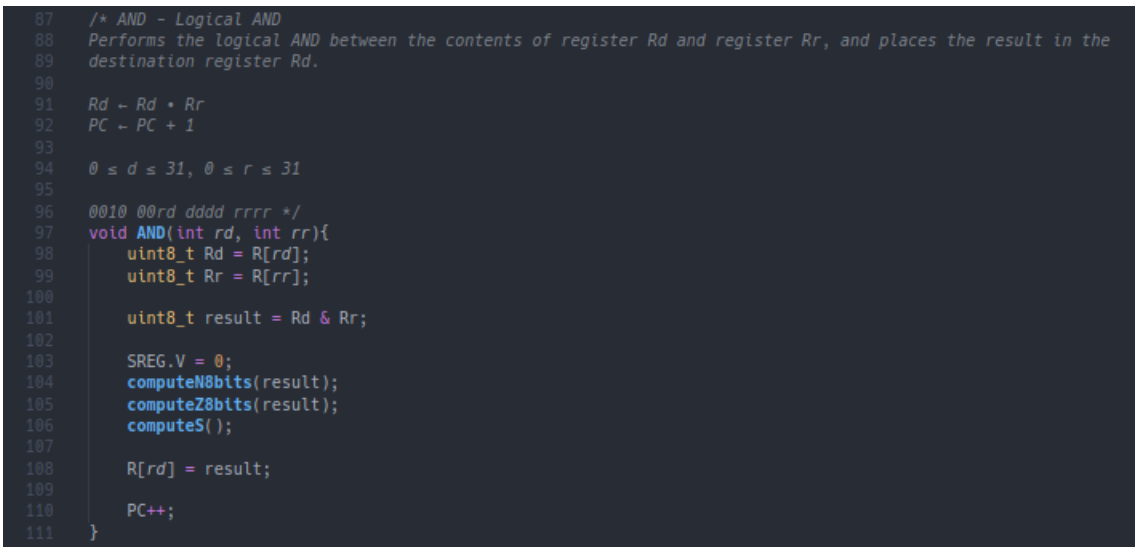

Figura 5. Instrução *AND* entre dois registradores.

3.3. Instruções de Saltos

Foram implementadas todas as funções de saltos disponíveis no conjunto de instruções do microcontrolador. Essas funções consistem apenas em alterar o valor do registrador PC (*Program Counter*) de acordo com uma dada condição.

4. Próximos Passos

Para incrementar a implementação do ambiente de simulação são sugeridos os seguintes passos:

- Implementação das instruções que estão faltando;
- Implementação dos registradores que controlam as portas digitais do microcontrolador (*PORTD, PORTB*);
- Implementação dos registradores que controlam a direção dos dados dos pinos como entrada ou saída (*DDRD, DDRB*);
- Implementação da memória onde será armazenado o código das instruções que serão executadas.

Referências

Atmel (2016). Avr instruction set manual.