

Atividade_01 - Revisão de Conceitos. Livro AVR e Arduino – Técnicas de Projeto
Capítulos: 1-Introdução e 2-O ATmega328

Obs.: Deve ser entregue arquivo contendo as perguntas e respectivas respostas.

1. Atualmente, os microcontroladores estão presentes em quase todos os dispositivos eletrônicos controlados digitalmente. Cite três exemplos destes equipamentos encontrados nas casas, nos veículos e nos eletrônicos portáteis. Obs.: Três exemplos de cada categoria.

Resposta:

Nas casas: Micro-ondas, máquina de lavar e telefone.

Nos veículos: Freio ABS, GPS e computador de bordo.

Nos eletrônicos portáteis: Celulares, videogames e relógios.

2. Quanto à organização do barramento, existem duas arquiteturas predominantes para as CPUs dos microprocessadores, a arquitetura Von-Neumann e a arquitetura Harvard. Explique as características de cada uma delas.

Resposta:

Arquitetura Von-Neumann:

A busca de dados e instruções não pode ser executada ao mesmo tempo. Apesar dessa limitação, a busca antecipada de instruções (pipeline) e caches de instruções podem ser uma solução.

Arquitetura Harvard:

Diferente da arquitetura Von-Neumann, os dados e instruções podem ser acessados simultaneamente.

3. Utilize a Figura 1.4 para explicar com um exemplo de código a diferença entre os Computadores com Conjunto Complexo de Instruções (CISC - Complex Instructions Set Computers) e Computadores com Conjunto Reduzido de Instruções (RISC – Reduced Instructions Set Computers).

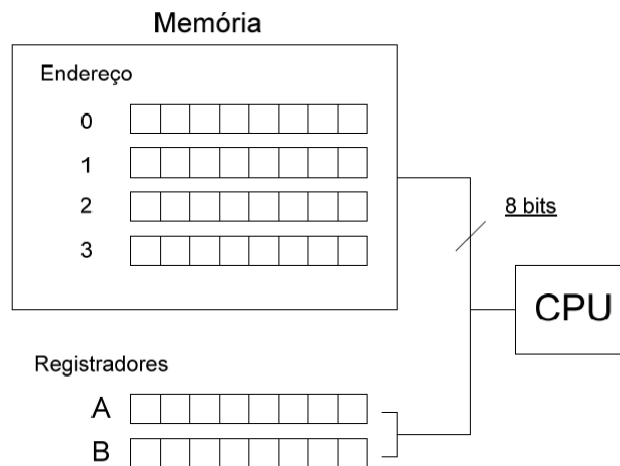


Fig. 1.4 – Diagrama esquemático para a comparação entre um microprocessador CISC e um RISC.

Resposta:

Um dos principais objetivos dos microprocessadores CISC, é executar a tarefa com o menor número de códigos possíveis (assembly). Em CISC's, o dado pode ser guardado sem a necessidade do uso explícito de registradores. Segue abaixo um exemplo de código de um microprocessador CISC:

MULT 0,3 //multiplica o conteúdo do endereço 0 com o do endereço 3
//armazena o resultado no endereço 0.

Para um microprocessador RISC, a resolução do problema poderia ser feita como:

LOAD A,0 //carrega o registrador A com o conteúdo do endereço 0
LOAD B,3 //carrega o registrador B com o conteúdo do endereço 3
MULT A,B //multiplica o conteúdo de A com o de B, o resultado fica em A
STORE 0,A //armazena o valor de A no endereço 0

O microprocessador RISC consiste na ideia de carga e armazenamento (Load and Store) utilizando registradores de uso geral. Antes dos dados serem multiplicados, necessitam, primeiramente, serem carregados em determinados registradores e, em seguida, o resultado será armazenado em uma posição de memória específica.

4. Defina o que é um microcontrolador e descreva as funcionalidades oferecidas por eles.

Resposta:

Um microcontrolador é o um sistema microprocessado com várias funcionalidades (periféricos) disponíveis em um único chip. Basicamente, um microcontrolador é um microprocessador com memórias de programa, de dados e RAM, temporizadores e circuitos de clock embutidos.

As funcionalidades encontradas nos microcontroladores são: gerador interno independente de clock (não necessita de cristal ou componentes externos), memória SRAM, EEPROM e flash, conversores analógicos-digitais (ADCs), conversores digitais-analógicos (DACs), vários temporizadores/contadores, comparadores analógicos, saídas PWM, além de diferentes tipos de interface de comunicação, incluindo USB, USART, I2C, CAN, SPI, JTAG e Ethernet, relógio de tempo real, circuitos para gerenciamento de energia no chip, circuitos para o controle de inicialização (reset), alguns tipos de sensores, interface para LCD, e entre outros.

5. Descreva sobre 10 das principais características do microcontrolador ATmega328.

Resposta:

- Microcontrolador de baixa potência, com arquitetura RISC avançada.
- 131 instruções, a maior parte executada em 1 ou 2 ciclos de clock (poucas em 3 ou 4 ciclos).
- 32 registradores de trabalho de propósito geral (8 bits cada). Alguns trabalham em par para endereçamentos de 16 bits.
- Operação de até 20 MIPS a 20 MHz.
- Multiplicação por hardware em 2 ciclos de clock.
- 32 kbytes de memória de programa flash de auto programação In-System (8 k, 16 k, nos respectivos ATmega88 e ATmega168).
- 1 kbytes de memória EEPROM
- 2 kbytes de memória SRAM.
- Ciclos de escrita e apagamento: memória flash 10 mil vezes, EEPROM 100 mil vezes.
- Seção opcional para código de boot para programação In-System por boot loader6.

6. Quais são os 11 periféricos do microcontrolador ATmega328?

Resposta:

- 23 entradas e saídas (I/Os) programáveis.
- 2 Temporizadores/Contadores de 8 bits com Prescaler separado, com modo de comparação.
- 1 Temporizador/Contador de 16 bits com Prescaler separado, com modo de comparação e captura.
- Contador de tempo real (com um cristal externo de 32,768 kHz conta precisamente 1 s).
- 6 canais PWM.
- 8 canais AD com resolução de 10 bits na versão TQFP (Thin profile plastic Quad Flat Package) e 6 canais na versão PDIP (Plastic Dual Inline Package)
- Interface serial para dois fios orientada a byte (TWI), compatível com o protocolo I2C.
- Interface serial USART.
- Interface serial SPI Master/Slave
- Watchdog Timer com oscilador interno separado.
- 1 comparador analógico.

7. O que é boot loader e onde está localizado na memória do microcontrolador Atmega328?

Resposta:

Boot loader é um pequeno programa que pode ser escrito no início ou no final da memória de programa e serve para que o microcontrolador gerencie a gravação de sua memória. Para tal, é necessária uma interface de comunicação externa com o software de desenvolvimento. No AVR o espaço destinado ao boot loader fica no final da memória de programa e a comunicação é feita através de um dos seus periféricos de comunicação, a USART.

8. Explique a arquitetura Harvard empregada pelo ATmega328 mostrada no diagrama da Figura 2.1.

- a) Barramento de dados.
- b) Barramento de instruções.
- c) Barramento de endereços.

Resposta:

O barramento de dados é de 8 bits, caracterizando o número de bits do microcontrolador. As instruções do ATmega são de 16 ou 32 bits, sendo em sua maioria de 16 bits. Assim, cada instrução consome 2 ou 4 bytes na memória de programa (um byte par e um ímpar). O acesso às posições de memória, fornecido pelo contador de programa (Program Counter - PC), é realizada de dois em dois bytes, começando sempre por uma posição par. Portanto, o barramento de endereços deve ser capaz de endereçar sempre posições pares da memória de programa. Diante disso, o bit menos significativo do barramento de endereços pode ser desprezado. Desta forma, para a memória de 32 kbytes do Atmega328 são necessários 14 bits de endereçamento ($2^{14} = 16.384$ endereços) e não 15 ($2^{15} = 32.768$ endereços).

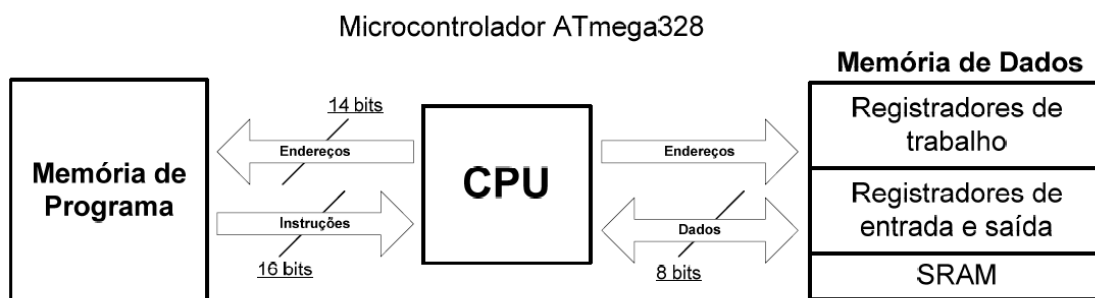


Fig. 2.1 – Diagrama esquemático da estrutura de um microcontrolador ATmega328.

9. Utilizando o diagrama de tempo mostrado na Figura 2.4, explique a técnica de Pipeline empregada na arquitetura do ATmega328.

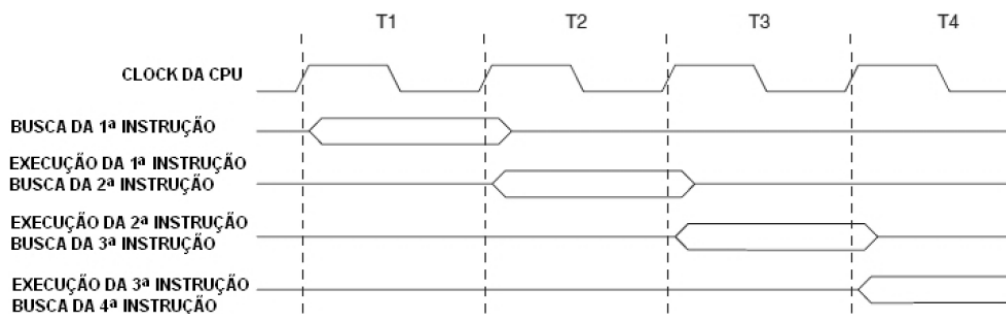


Fig. 2.4 – Diagrama de tempo para a busca e execução de instruções no ATmega.

Resposta:

Pipeline é uma técnica de implementação de processadores que permite a sobreposição temporária das diversas fases de execução das instruções. Ela aumenta o número de instruções executadas simultaneamente e a taxa de instruções iniciadas e terminadas por unidade de tempo.

O ATmega emprega um pipeline de 2 estágios, dessa forma uma instrução é lida e decodificada dentro de um mesmo ciclo de clock, enquanto a instrução anterior é executada.

10. Utilizando a Figura 2.5, explique a organização das memórias de dados e SRAM do ATmega328.

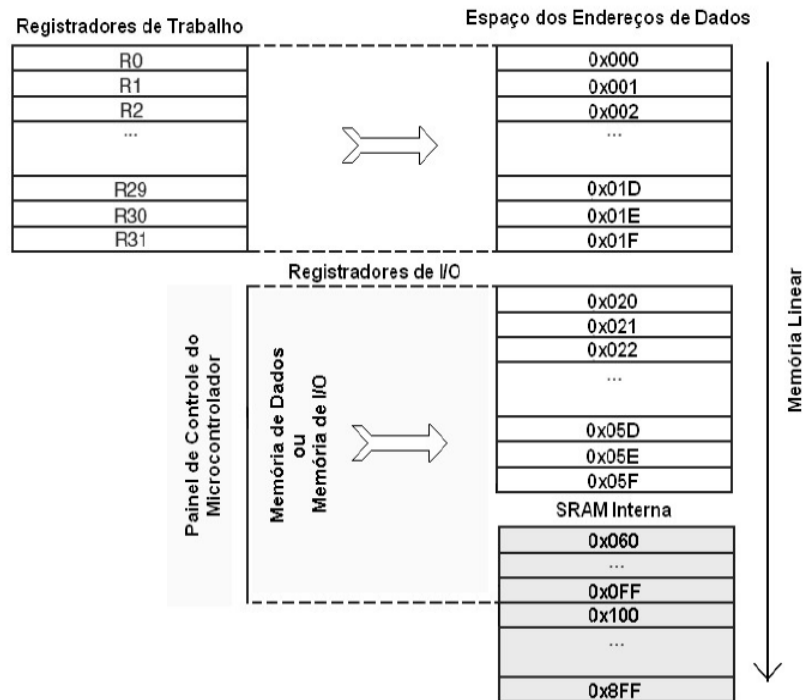


Fig. 2.5 – Memória de dados e memória SRAM do ATmega328.

Resposta:

A organização da memória é linear, sendo seu endereço inicial 0(zero) seguindo até o endereço 2303(0x8FF). Das posições de memória, 32 pertencem aos registradores de uso geral (0x000 até 0x01F), 64 aos registradores de entrada e saída (0x020 até 0x05F) e 2048 bytes pertencem à memória SRAM (0x060 até 0x8FF), sendo que 160 primeiros endereços são empregados para a extensão dos registradores de entrada e saída. Isso é necessário porque o número de periféricos no ATmega328 é superior ao que pode ser suportado pelos 64 registradores originais (dos primeiros ATmegas) além de permitir o acréscimo de futuras 23 funcionalidades. Desse modo, é utilizado a memória SRAM7 para aumentar o número de registradores de I/O sem que haja a necessidade de maiores modificações no projeto do chip.

11. Por que os registradores de I/O são chamados de “painel de controle” do microcontrolador?

Resposta:

Os registradores de I/O são chamados de “painel de controle” do microcontrolador, pois todas as configurações de trabalho se encontram nessa parte da memória, inclusive o acesso às entradas e saídas. São esses registradores que dão acesso às funcionalidades do microcontrolador, pois possuem todas as informações relacionadas aos periféricos e ao processamento da CPU. É com esses registradores que o desenvolvedor se familiarizará ao trabalhar com os periféricos (dita como as “chaves” que ligam e desligam tudo).

12. Explique o que é um PORT e como eles estão organizados no microcontrolador ATmega328.

Resposta:

Os pinos do ATmega são organizados em conjuntos denominados: PORTB, PORTC e PORTD. Cada um deles possui 8 pinos (com exceção do PORTC) organizados pelos nomes: PB0-7, PC0-6 e PD0-7. Cada PORT é uma porta bidirecional de I/O de 8 bits com resistores

internos de pull-up selecionáveis para cada bit. Os registradores de saída possuem características simétricas com capacidade de fornecer e receber corrente, suficiente para acionar cargas de até 40 mA. Outra característica importante é que todos os pinos apresentam pelo menos duas funções distintas, até mesmo o pino de reset pode ser utilizado como pino de I/O.

13. O AVR suporta várias opções de clock, identifique qual é a opção de clock e a frequência utilizada na placa Arduino.

Resposta:

Para a redução do consumo de potência, os módulos de clock podem ser suspensos usando diferentes modos de programação. O AVR suporta as seguintes opções de clock: cristal ou ressonador cerâmico externo, cristal de baixa frequência externo, sinal de clock externo e oscilador RC interno.

Quando o uso de um cristal ou ressonador cerâmico externo não é necessário, pode-se utilizar o oscilador interno, o qual permite o uso de pinos (XTAL1 e XTAL2) do ATmega328 para outras funções. O oscilador interno pode ser programado para operar com a frequência máxima de 8 MHz (o valor default é 1 MHz). Apesar de ter sua frequência dependente da tensão e da temperatura, a frequência do oscilador interno pode ser precisamente calibrada pelo usuário.

14. Descreva sobre as quatro fontes de RESET do ATmega328.

Resposta:

- Power-on Reset: ocorre na energização enquanto a fonte de alimentação estiver abaixo da tensão limiar de power-on reset (VPOT).
- Reset externo: ocorre quando o pino de reset é aterrado (0 V) por um determinado período de tempo.
- Watchdog Reset: ocorre quando o watchdog está habilitado e o seu contador atinge o valor limite.
- Brown-out Reset: ocorre quando a tensão de alimentação cair abaixo do valor definido para o brown-out reset (VBOT) e o seu detector estiver habilitado.

15. Utilizando a Figura 2.6, explique a organização da memória de programa do ATmega328.

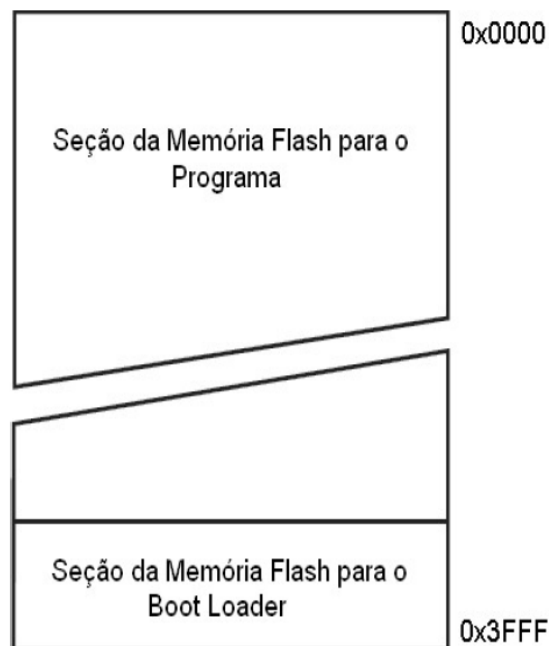


Fig. 2.6 – Organização da memória de programa.

Resposta:

De acordo com a figura da organização da memória de programa fornecida pelo enunciado, cada endereço da memória possui 2 bytes, pois as instruções do AVR são de 16 ou 32 bits. Dessa forma, a memória possui um total de 16384 endereços (de 0x0000 até 0x3FFF), correspondendo a 32 kbytes de memória. Existe uma seção específica para carregar o boot loader, que pode ou não ser utilizada para esse fim. A memória flash suporta, no mínimo, 10 mil ciclos de escrita e apagamento. A memória EEPROM é de 1 kbytes e é organizada separadamente. Cada byte individual pode ser lido ou escrito e a memória suporta, no mínimo, 100 mil ciclos de escrita e apagamento.

16. Utilizando a Figura 2.7, explique o funcionamento do Stack Pointer do ATmega328.

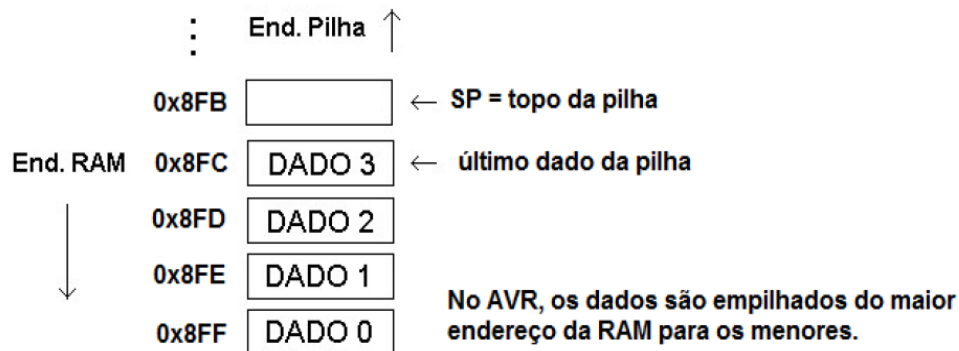


Fig. 2.7 – Stack Pointer.

Resposta:

O Stack Pointer (SP), traduzido como ponteiro de pilha, é um registrador que armazena um endereço correspondente a uma posição da memória RAM, em que é utilizada na forma de uma pilha para armazenamento temporário de dados: variáveis locais e endereços de retorno após chamadas de sub-rotinas e interrupções. Resumidamente, o SP indica a posição onde um determinado dado foi armazenado na pilha alocada na RAM.

O endereço do ponteiro de pilha é pós-decrementado toda vez que um dado é colocado na pilha. Assim, a cada novo dado colocado na pilha, o endereço do SP apresenta um valor menor que o anterior. Igualmente, quando um dado é retirado da pilha, o endereço do SP é pré-incrementado, sempre apontando uma posição acima do último dado válido da pilha. Isso implica que o comando PUSH, que coloca um dado na pilha, diminui o valor do SP e o comando POP, que retira um dado da pilha, o incrementa.

Dado o funcionamento do SP, na sua inicialização ele deve apontar para o endereço final da RAM, no caso do ATmega328 o endereço é 0x8FF (seu valor default após a energização). Entretanto, existem microcontroladores da família ATmega que precisam ter o SP inicializado pelo programador. Nesse caso, quando o programa é escrito em assembly a inicialização deve ser feita pelo programador de forma explícita. Essa inicialização é realizada de forma automática quando o programa é escrito na linguagem C, pois o compilador se encarrega da tarefa.

Como o SP armazena um endereço da RAM, ele deve ter um número de bits suficiente para isso. Como o ATmega possui registradores de 8 bits, são necessários dois registradores para o SP, um armazena a parte baixa do endereço (SP Low) e outro armazena a parte alta do endereço (SP High), resultando num registrador de 12 bits (os 4 bits mais significativos do SPH não são utilizados).

Parte prática

Título: Conhecendo o Tinkercad para simulação de circuitos

Objetivos: Familiarização com o Tinkercad para a simulação de circuitos simples.

Nesta prática utilizaremos o Tinkercad para simular um circuito simples usando o microcontrolador Atmega328p, utilizado nas placas Arduino UNO.

Procedimentos:

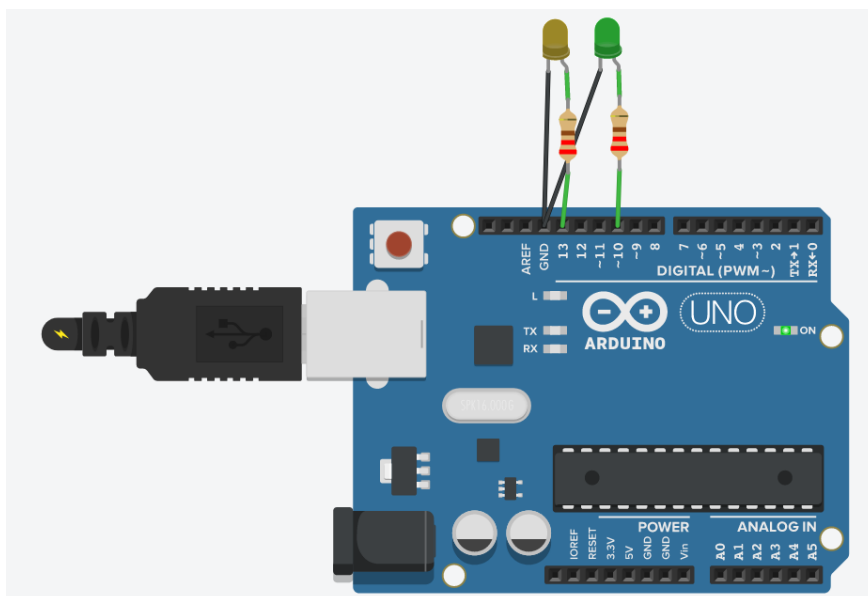
1. Crie uma conta no Tinkercad, caso não possua ([tinkercad.com](https://www.tinkercad.com)).
2. Em seguida, vá para a aba circuits (<https://www.tinkercad.com/circuits>).
3. Você deve fazer um circuito capaz de piscar um led. Note que este projeto já está disponível (na aba Starters → Arduino).
4. Modifique o projeto de forma a provocar flashes intermitentes. O led deve ficar apagado por 500ms e aceso por apenas 50ms.
5. Adicione um segundo led que acende na sequencia do primeiro. Assim, a sequencia de ativação seria: LED1 (50ms), LED2(50ms), 450ms, LED1 (50ms), ...
6. Use um led amarelo para o LED1 e verde para o LED2.
7. Cole o código fonte do microcontrolador ao final deste arquivo e inclua a imagem de seu design.

```
void setup()
{
  DDRB = 0b00100000;
  DDRB |= 1<<PB2;
}

void loop()
{
  PORTB = 0b00100000;
  delay(50);

  PORTB = 0b00000100;
  delay(50);

  PORTB = 0;
  delay(450);
}
```



Rúbrica:

Questões 01 a 11: 4% cada

Questões 12 a 16: 7% cada

Prática: Código coerente e funcional: 10%, design do circuito: 4%

Valor desta atividade na média: 0.5