

Atividade_02 - Revisão de Conceitos. Livro AVR e Arduino – Técnicas de Projeto Capítulos: 4 (Manipulação de bits) e 5 (Portas)

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

Título: Conhecendo o Atmega328 e acessando portas de saída usando C

Objetivos: Aprender a consultar a folha de dados (datasheet) do microcontrolador em busca de informações sobre o mesmo.

Nesta prática utilizaremos o Tinkercad para simular um circuito simples usando o microcontrolador Atmega328, utilizado nas placas Arduino UNO. Desta vez, programaremos usando um código C com as diretivas/macros definidos pela AVR para acesso aos registradores.

1. Perguntas teóricas (consultar datasheet)

Utilizando a folha de dados do microcontrolador (uC) Atmega328, da Atmel, responda as questões abaixo relacionadas:

1. As instruções nestes uC tem sempre o mesmo número de parâmetros? Exemplifique.

R:

Não, a quantidade de parâmetros depende da instrução que você está executando, por exemplo as instruções mul e mov abaixo têm quantidades distintas de params:

```
mov ah, 09h;  
mul 5;
```

2. As instruções têm sempre o mesmo tamanho em bits? Exemplifique.

R:

Não, enquanto as instruções AVR possuem 16 bits, as instruções de endereçamento direto, LDS e STS, ocupam 32 bits.

3. Qual(is) instrução(ões) pode(m) ser usada(s) para realizar uma multiplicação neste uC (não citar as iniciadas por F)? Quais são os operandos? Onde os resultados são armazenados?

R:

As instruções que podem ser usadas para multiplicação neste uC são:

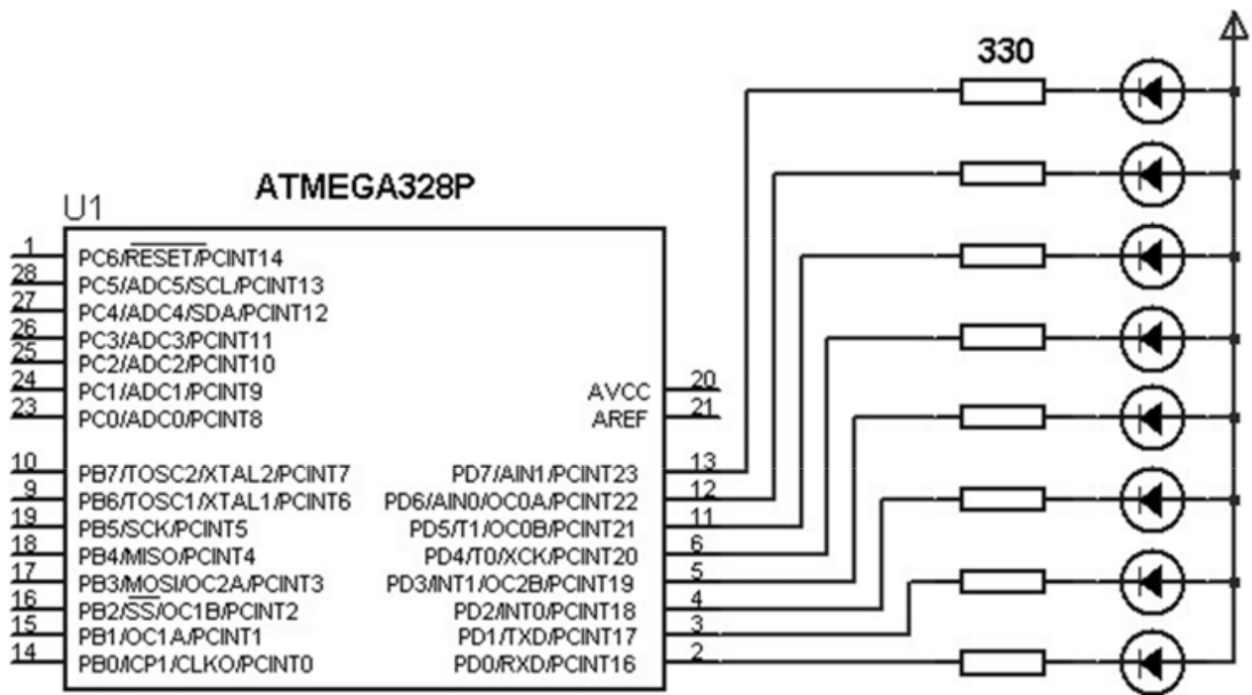
```
MUL (Multiply Unsigned), operandos Rd e Rr;  
MULS (Multiply Signed), operands Rd e Rr;  
MULSU (Multiply Signed with Unsigned), operands Rd e Rr
```

O local de armazenamento dos resultados é no par de registradores DX:AX

2. Procedimentos práticos (ler “Manipulando bits”):

1. Acesse sua conta no Tinkercad ([tinkercad.com](https://www.tinkercad.com)) e vá para a aba circuits (<https://www.tinkercad.com/circuits>).

2. Desenhe o circuito que segue no Tinkercad.



3. Utilizando o deslocamento de bits crie um programa em C que ligue 8 LEDs conforme a figura, da seguinte forma:

- Ligue sequencialmente 1 LED da direita para a esquerda (o LED deve permanecer ligado até que todos os 8 estejam ligados, depois eles devem ser desligados e o processo repetido).
- Ligue sequencialmente 1 LED da esquerda para a direita, mesma lógica da letra ^a
- Ligue sequencialmente 1 LED da direita para a esquerda, desta vez somente um LED deve ser ligado por vez.
- Ligue sequencialmente 1 LED da esquerda para a direita e vice-versa (vai e volta), só um LED deve ser ligado por vez.

* Deixe um intervalo de 300ms entre cada troca na ativação dos LEDs.

4. Cole o código fonte do microcontrolador ao final deste arquivo e inclua a imagem de seu design. Importante: Deixe seu circuito público no Tinkercad e cole o link para ele aqui:

https://www.tinkercad.com/things/b4tcmDSfWpv-magnificent-snicket-leelo/editel?sharecode=U8f_Ug8ZxUxlaaYlQrB3Z41DoK-A0KdqT6n1cKtXfPA


```

    }
    b = 0;
    PORTD = ~b;
    delay(t);
}

//dir p esq 1 a 1
for(int i=0; i<2; i++){
    for(int i=0; i<8; i++){
        b = 1 << i;
        PORTD = ~b;
        delay(t);
    }
    b = 0;
    PORTD = ~b;
    delay(t);
}

//dir p esq 1 a 1 e vice e versa
for(int i=0; i<2; i++){

    for(int i=0; i<=7; i++){ //dir p esq
        b = 1 << i;
        PORTD = ~b;
        if(i != 7) delay(t); //impedir delay no ultimo led
    }

    for(int i=7; i>=0; i--){ //esq p dir
        b = 1 << i;
        PORTD = ~b;
        if(i != 0) delay(t); //impedir delay no ultimo led
    }

}
b = 0;
PORTD = ~b;
delay(t);
}

```

ATENÇÃO: Usar as funções `pinMode()` e `digitalWrite()` estão proibidos nesta prática. O uso delas fará a nota atribuída ser zero.

ATENÇÃO: Documente seu código. Cada linha/bloco deve deixar explícito o seu papel.

Rúbrica:

Perguntas teóricas: 1 a 3: 4% cada

Procedimentos práticos:

3a a 3d: 19% cada

Desenho do circuito: 12%

Valor desta atividade na média: 0.7