

## Questões Teórica

1) Qual o valor ao final do registrador **AUX** após a execução dos seguintes trechos em Assembly-AVR executados em um microcontrolador ATMEGA328P **(1,6)**

a) .def aux = r16 LDI R16,0b11111111 LDI R17,0b11110000 EOR R16, R17	b) .def aux = r16 LDI R16, 0b00001111 INC R16 CLR R17 LDI R17,10 ADD R16,R17
c) .def aux = r16 LDI R17,0b00000011 OUT PORTB,R17 SBIC PORTB,1 RJMP PASSO_1 PASSO_0: LDI R16,15 RJMP FIM PASSO_1: LDI R16,20 RJMP FIM FIM:	d) .def aux = r16 LDI R16,5 LDI R17,0b00000111 LOOP: DEC r16 EOR R16,R17 BRNE LOOP RJMP PASSO_0 PASSO_0: LDI R17,15 RJMP FIM PASSO_1: LDI R17,20 RJMP FIM FIM:

2) Coloque V (verdadeiro) ou F (falso). Em relação aos elementos internos constituintes de um sistema microcontrolado considerando AVR Atmega 328P. **(1,5)**

- ☐ A instrução em Assembly LDI tem efeito sobre o registrador PORTB
- ☐ A instrução em Assembly SBIC tem efeito sobre o registrador R16.
- ☐ Ao mover o valor 0x0F para o registrador DDRB, o pino RB1 estará configurado como entrada.
- ☐ Ao mover o valor 0x00 para o registrador PINB, o pino RB0 estará configurado como entrada.
- ☐ A memória EEPROM dos microcontroladores é um recurso de memorização dos dados não voláteis para o programa de um microcontrolador.
- ☐ A arquitetura RISC sempre será uma configuração de barramentos de dados e instruções do tipo Harvard.

☐ Assembler é uma linguagem de máquina e Assembly é um o programa compilador.

☐ A arquitetura RISC possui uma configuração de barramentos de dados e programa compartilhados.

☐ A memória RAM (Random Access Memory) é utilizada para armazenar dados do programa.

☐ O microcontrolador AVR Atmega328P possui uma pilha com 31 endereços de capacidade.

☐ Os pinos de cada porta precisam ser configurados, individualmente, como entrada ou saída, através de registradores específicos PINB, PINC e PIND.

☐ Microcontroladores de arquitetura Harvard executam instruções CISC de forma mais rápida comparada a arquitetura Von Neumann.

☐ O programa do usuário armazenado em uma memória volátil de tecnologia FLASH, sendo cada instrução ocupa 2 bytes da memória de programa de 32 Kbytes

☐ O recurso PIPELINE consiste em executar e buscar a próxima instrução em apenas um ciclo de clock.

☐ Os registradores de Uso Geral (GPR) são utilizados para acionar funções específicas e periféricos internos.

3) Desenhe os blocos básicos de funcionamento para um microcontrolador Atmega 328P **(0,9)**

4) Elabore o trecho inicial de um programa assembly, considere a seguinte figura para definir corretamente os registradores I/O. Considere a ativação pull-up para entradas. **(1,0)**

	30	<input type="checkbox"/>	AVCC
OUTPUT (1)	29	<input type="checkbox"/>	PC7 (TOSC2)
INPUT (0)	28	<input type="checkbox"/>	PC6 (TOSC1)
OUTPUT (1)	27	<input type="checkbox"/>	PC5 (TDI)
OUTPUT (1)	26	<input type="checkbox"/>	PC4 (TDO)
INPUT (0)	25	<input type="checkbox"/>	PC3 (TMS)
INPUT (0)	24	<input type="checkbox"/>	PC2 (TCK)
INPUT (0)	23	<input type="checkbox"/>	PC1 (SDA)
OUTPUT (1)	22	<input type="checkbox"/>	PC0 (SCL)
	21	<input type="checkbox"/>	PD7 (OC2)

Uma determinada fábrica de estampo possui uma máquina controlada por um CLP ilustrada na figura. O setor de manutenção da empresa solicitou a um programador para que o mesmo substitua o CLP por um microcontrolador ATMEGA328P. Elabore as rotinas do programa em Assembly-AVR que atenda ao projeto solicitado. (5,0)

**Defina pinos de entrada e saídas (PORTB E PORTD) e faça uma legenda para cada entrada e saída utilizada.**

Utilize máquina de estados ou sequencia de eventos para sua programação.

Um display de sete segmentos indica o estado atual do ciclo para monitoramento do usuário.

Ao ligar a máquina, esta aguarda até que o botão START seja pressionado; [Display=0]

Ao pressionar o botão Start, a válvula V4 será fechado (HIGH), a válvula V1 abrirá até que o sensor nível 1 detecte a presença de líquido; [Display=1]

Em seguida a válvula V2 será ligada por 3 segundos; [Display=2]

Em seguida a válvula V3 será ligada por 3 segundos; [Display=3]

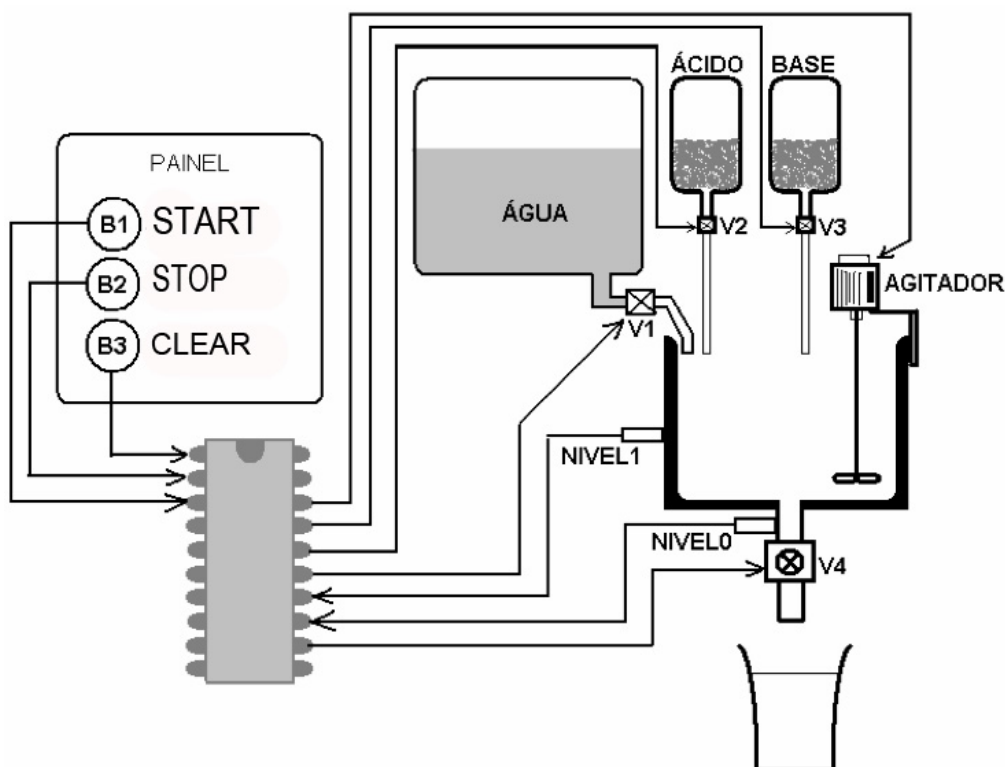
após este tempo o agitador é acionado durante 2 segundos. [Display=4]

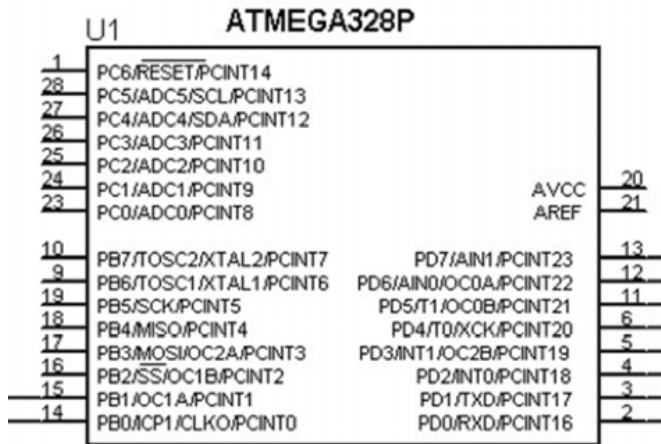
Ao final V4 deverá ser aberto até que o sensor nível 0 fique em nível LOW retornando ao estado inicial; [Display=5]

Considere o botão “STOP” conectado ao Reset do microcontrolador e realiza a parada em qualquer instante, sendo que ao resetar ou início do processo todas as saídas estão desligadas.

O botão CLEAR não será utilizado.

Considere o uso de clock de 16MHz.





```
.equ SP1 = pb0
.equ AL = pb4
.equ M1 = pb5
.def caixas = r0
.ORG 0x00
include "biblioteca.h"
    rjmp Start
Start:
    cbi ddrb,0
    sbi portb,0
    sbi ddrb,4
    sbi ddrb,5
    ldi aux,0
    mov caixas,aux
    mov display_number,caixas
    rcall display_write
Loop:
    sbi portb,M1
    cbi portb,AL
SP1ON:
    sbic pinb,SP1
    rjmp SP1ON
    ldi delay_time,1
    rcall delay_seconds
SP1OFF:
    sbis pinb,SP1
    rjmp SP1OFF
    inc caixas
    mov display_number,caixas
    rcall display_write
    ldi aux,9
    eor aux,caixas
    brne Loop
    breq ALERTA
    rjmp Loop
ALERTA:
    cbi portb,M1
    sbi portb,AL
    ldi delay_time,1
    rcall delay_seconds
    cbi portb,AL
    ldi delay_time,1
    rcall delay_seconds
    rjmp ALERTA
```

```
# biblioteca.h
.def delay_time = r25
.def display_number = r24
.def aux = r16
delay_seconds:
    ldi r31,82
    ldi r30,0
    ldi r29,0
loop_delay:
    dec r29
    brne loop_delay
    dec r30
    brne loop_delay
    dec r31
    brne loop_delay
    dec delay_time
    brne delay_seconds
    ret

///// biblioteca de decodificacao do DISPLAY
.equ DISPLAY = PORTD
LDI R16,0b11111011
OUT DDRD, R16 ;PORTD pin2 entrada, demais saídas
LDI R16,0b11111111
OUT PORTD, R16 ;desliga o display, pull up em pin2
;-----
;SUB-ROTINA que decodifica um valor de 0 -> 15 para o display
;-----
Decodifica:
    LDI ZH,HIGH(Tabela<<1)
    LDI ZL,LOW(Tabela<<1)
    CLR R1
    ADD ZL,AUX
    ADC ZH,R1
    LPM R24,Z
    OUT DISPLAY,R24 ;mostra no display
    RET

;-----
Tabela: .dw 0xF584, 0x644C, 0x2635, 0xF406, 0x3404, 0x0714, 0x458E, 0x1E0E
;-----
           1 0    3 2    5 4    7 6    9 8    B A    D C    F E
```