 INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA SÃO PAULO	Curso: Eng. de Controle e Automação	Data: 07/10/25	Nota
	Período: Integral	Prova 1/2	
	Disciplina: Microprocessadores e Microcontroladores	Peso: 30%	
	Prof.: Marcos Aparecido Chaves Ferreira	Revisão do Aluno:	
Aluno (a):			

Questões sem consulta

1) Qual o valor ao final do registrador **AUX** após a execução dos seguintes trechos em Assembly-AVR executados em um microcontrolador ATMEGA328P **(1,5)**

a) .def aux =R16 LDI R16,0b111110101 LDI R17,0b111110000 EOR R16, R17 ADD R16,R17	b) .def aux = R16 LDI R17, 0b00000111 DEC R16 STS 0x100,R16 SUB R16,R17
c) .def aux =R16 LDI R16,0b00000011 OUT DDRB,R16 OUT PORTB,R16 SBIC PORTB,1 RJMP PASSO_1 PASSO_0: LDI R16,20 RJMP FIM PASSO_1: LDI R16,21 RJMP FIM FIM:	d) .def aux = r16 LDI R16,11 LDI R17,0b00000111 LOOP: DEC R16 CP R16,R17 BRNE LOOP PASSO_0: LDI R17,14 RJMP FIM PASSO_1: LDI R17,18 RJMP FIM FIM:

Obs: BRNE label ; desvia quando Z=0 (operação não zerada).

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

- () A memória EEPROM é um recurso de memorização volátil para o programa de um microcontrolador.
- () O AVR Atmega328P possui 32 registradores com função específica entre R0 e R31.
- () Os pinos de entrada são configurados com registradores PINx e saídas através de registradores OUTx.
- () Microcontroladores de arquitetura Harvard executam instruções CISC de forma mais rápida comparada a arquitetura Von Neumann.
- () O recurso de pipeline consiste em executar apenas uma instrução na CPU.
- () Os registradores de GPR são utilizados para acionar funções específicas e periféricos internos.
- () A instrução em Assembly LDI tem efeito sobre o registrador DDRB
- () A instrução em Assembly SBIC tem efeito

sobre o registrador R16.

() O registrador PC armazena o valor lido de uma porta digital, para que a próxima instrução saiba qual pino será utilizado.

() Ao mover o valor 0xF0 para o registrador DDRB, o pino PB0 estará configurado como saída.

() Ao mover o valor 0xF0 para o registrador PINB, o pino PB1 estará configurado como entrada.

() A arquitetura RISC sempre terá instruções do tipo Harvard

() A arquitetura CISC possui uma configuração de barramentos de dados e programa compartilhados.

() A memória RAM (Random Access Memory) é utilizada para armazenar dados não voláteis para o programa.

() A ULA do ATmega controla diretamente os pinos das portas B, C e D.


3) Elabore o trecho inicial de um programa Assembly. **Altere** a figura a seguir e **defina** corretamente os registradores de I/O e respectivos pinos para o PORTC de um ATMEGA. Um botão com pull-up conectado ao pino PC0, ao ser pressionado, é curto-circuitado ao GND. Enquanto pressionado, os pinos de PC4 até PC7 devem piscar em sequência alternando 1 segundo; caso contrário, permanecem em LOW **(2,0)**.

	30	□	AVCC
OUTPUT (1)	29	□	PC7 (TOSC2)
INPUT (0)	28	□	PC6 (TOSC1)
OUTPUT (1)	27	□	PC5 (TDI)
OUTPUT (1)	26	□	PC4 (TDO)
INPUT (0)	25	□	PC3 (TMS)
INPUT (0)	24	□	PC2 (TCK)
INPUT (0)	23	□	PC1 (SDA)
OUTPUT (1)	22	□	PC0 (SCL)
	21	□	PD7 (OC2)

```

cbi ddrb,0
sbi portb,0
loop:
  sbis pind,0
  rjmp apaga
  rjmp acende
acende:
  ldi r16,0b01010000
  out portb,R16
  ldi delay_time,1
  rcall delay_seconds
apaga:
  clr r16
  rjmp loop

```

 INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA SÃO PAULO	Curso: Eng. Controle e Automação	Data: 07/10/2025	Nota
	Período: Integral	Prova 2/2	
	Disciplina: Micro. Microcontroladores	Peso: 30%	
	Prof.: Marcos Aparecido Chaves Ferreira	Revisão do Aluno:	
Aluno (a):			

Projeto com consulta

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 PORTC) e faça uma legenda para cada entrada e saída utilizada. A porta D esta conectada ao LCD

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

Um display lcd indica o estado atual de cada ciclo para monitoramento para o usuário.

Ao ligar a máquina, esta aguarda até que o botão START seja pressionado; [Aguardando Start] **(0,5)**

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; [V1 ON] **(0,5)**

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

Em seguida a válvula V2 será ligada por 2 segundos; [V2 ON] **(0,5)**

após este tempo o agitador é acionado durante 2 segundos. [AGITADOR ON] **(0,5)**

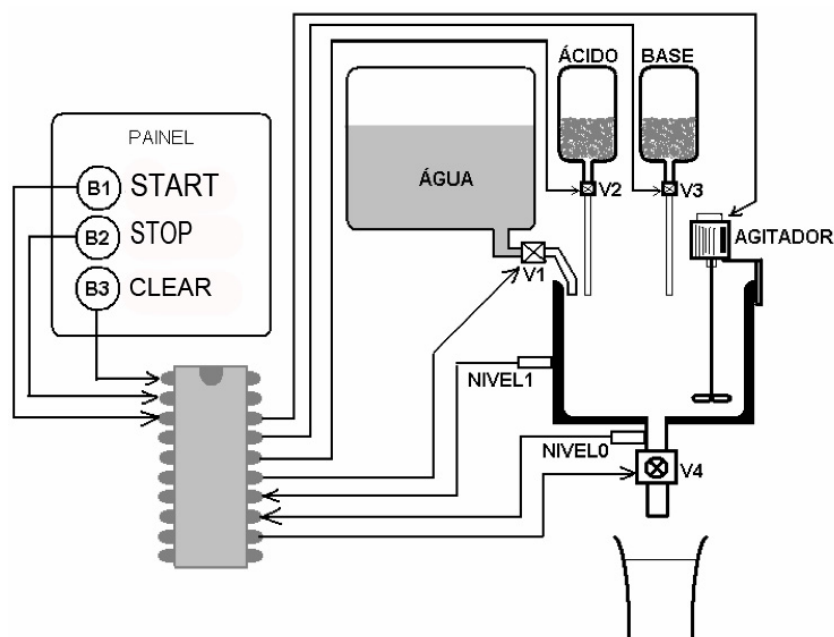
Ao final V4 deverá ser aberto até que o sensor nível 0 fique em nível LOW retornando ao estado inicial; [Esvaziando] **(0,5)**

Utilize interrupção INT0 para a função de [Parada Emergencial] e realiza a parada em qualquer instante, sendo desligando todas as saídas e retornando ao início do programa. **(1,5)**

O botão CLEAR não será utilizado.

Utilizando uma saída disponível, pisque um led a cada 1 segundo utilizando interrupção timer 0 enquanto em funcionamento. **(1,0)**

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 r16,0b01010000
    out TCCR0A,r16
    ldi r16,0b00000001
    sts TIMSK0,r16
    ldi r16,0b00000101
    out TCCR0B,r16
    ldi r16,100
    out TCNT0,r16
    sei
    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 display lcd
rcall lcd_init
rcall lcd_clear ;
ldi lcd_col,3 ;define coluna3
rcall lcd_lin0_col ;define linha 0
ldi lcd_caracter,'O'
rcall lcd_write_caracter
ldi lcd_caracter,'I'
rcall lcd_write_caracter

TIM0_OV:
    ldi r16,100
    out TCNT0,r16
    inc r17
    ldi r18,50
    cp r17,r18
    breq alterna
    reti
alterna:
    clr r17
    sbic portb,1
    rjmp apagaled
    sbi portb,1
    reti
apagaled: ; pisca o amarelo
    cbi portb,1
    reti

```