

O protocolo de comunicação I²C tem quantas linhas de dados? Quais?

Apenas uma, sendo o STA

b) Qual a velocidade do I²C?

Os barramentos I²C tem velocidade de 100 kbits/s no modo padrão e 10kbits/s no modo de baixa velocidade, mas arbitrariamente frequências baixas de clock também são permitidas. São velocidades acordadas entre o mestre e o escravo.

c) O que é Over Head?

É o tempo em que o sistema perde mandando ordens de controle ao invés de dados úteis.

d) Qual é o tamanho de Buffer do arduino para transmitir um dado usando um protocolo I²C?

Para o Arduino Uno é 32 bytes.

e) O que é registrador Pull Up ?

O registrador Pull Up é utilizado para evitar falhas de comunicação e é utilizado para evitar que um dispositivo se comunique com erros.

f) A gravação de dados é feita no clock de subida ou descida?

É feita no clock de subida.

g) Qual é o método de controle do protocolo UART?

É uma comunicação assíncrona, não há barramentos de clock para sincronia de dados enviados e recebidos pelo mestre e escravo.

h) Quais são os bits de controle do protocolo UART?

São o primeiro e último bit para realizar o controle.

i) Compare a ordem de dados em relação ao UART e I²C?

O UART a ordem de dados é invertida com o I²C, LSB deve ser lido primeiro no UART.

j) Explique o funcionamento da paridade ímpar e par no protocolo de comunicação UART?

No bit de paridade é inserido entre o fim dos bits de dados e o bit final. O valor do bit de paridade depende quando é usado ímpar e par.

8 bits de dados, um bit de parada, sem paridade 7 bits de dados, uma parada, paridade ímpar

l) Como funciona o slave select no SPI?

O sinal de SS da SPI funciona como Seleção de Escravo (Slave Select). É um sinal ativo em nível baixo, o que significa que o dispositivo é selecionado quando este pino se encontra em nível baixo.

m) Quem define a velocidade de comunicação no protocolo de comunicação SPI?

O mestre, basta que o escravo consiga se comunicar nessa velocidade.

Dec / Bin/ Hex

0/ 0000 / 0

1/ 0001 / 1

2/ 0010 / 2

3/ 0011 / 3

4/ 0100 / 4

5/ 0101 / 5

6/ 0110 / 6

7/ 0111 / 7

8/ 1000 / 8

9/ 1001 / 9

10/ 1010 / A

11/ 1011 / B









12/ 1100 / C

13/ 1101 / D

14/ 1110 / E

15/ 1111 / F

Portas lógicas

Função lógica	Símbolo lógico	Tabela verdade	Expressão booleana															
Porta Buffer	A  Y	<table><tr><th>A</th><th>Y</th></tr><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td></tr></table>	A	Y	0	0	1	1	$Y = A$									
A	Y																	
0	0																	
1	1																	
Porta NOT - Inversora	A  Y	<table><tr><th>A</th><th>Y</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	A	Y	0	1	1	0	$Y = \overline{A}$									
A	Y																	
0	1																	
1	0																	
Porta AND	A  B Y	<table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	Y	0	0	0	0	1	0	1	0	0	1	1	1	$Y = A \cdot B$
A	B	Y																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
Porta NAND	A  B Y	<table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	Y	0	0	1	0	1	1	1	0	1	1	1	0	$Y = \overline{A \cdot B}$
A	B	Y																
0	0	1																
0	1	1																
1	0	1																
1	1	0																
Porta OR	A  B Y	<table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	Y	0	0	0	0	1	1	1	0	1	1	1	1	$Y = A + B$
A	B	Y																
0	0	0																
0	1	1																
1	0	1																
1	1	1																
Porta NOR	A  B Y	<table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	Y	0	0	1	0	1	0	1	0	0	1	1	0	$Y = \overline{A + B}$
A	B	Y																
0	0	1																
0	1	0																
1	0	0																
1	1	0																
Porta XOR	A  B Y	<table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	Y	0	0	0	0	1	1	1	0	1	1	1	0	$Y = A \oplus B$
A	B	Y																
0	0	0																
0	1	1																
1	0	1																
1	1	0																
Porta XNOR	A  B Y	<table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	Y	0	0	1	0	1	0	1	0	0	1	1	1	$Y = \overline{A \oplus B}$
A	B	Y																
0	0	1																
0	1	0																
1	0	0																
1	1	1																

Diferença de Von Neumann para Harvard?

Von Neumann é uma arquitetura de computador digital cujo design é baseado no conceito de computadores de programa armazenados onde os dados do programa e os dados de instrução são armazenados na mesma memória. Esta arquitetura foi projetada pelo famoso matemático e físico John Von Neumann em 1945.

Para Harvard é a arquitetura de computador digital cujo design é baseado no conceito onde há armazenamento separado e barramentos separados (caminho de sinal) para instrução e dados. Foi desenvolvido basicamente para superar o gargalo da Arquitetura Von Neumann.

Lei de Moore

É uma observação e projeção de uma tendência histórica relacionado a Indústria de microchips e processamento de computadores. Foi observada por Gordon E. Moore, e consiste no estudo de que o número de transistores dos chips teria um aumento de 100%, pelo mesmo custo, a cada período de 18 meses.

Tipos de memórias

RAM - "Random Access Memory": memória de leitura/escrita para armazenamento

temporário de programas e dados, volátil;

RAM Estática - RAM com menor densidade e mais rápida que a RAM dinâmica.

Não necessita de circuitos adicionais em um microcomputador.

RAM Dinâmica - RAM com maior densidade e mais lenta que a RAM estática.

Necessita de circuitos adicionais de controle em um microcomputador.
ROM - "Read Only Memory": memória programada quando o chip é fabricado, não podendo ser modificada. É usada para armazenamento permanente de programas e dados;

PROM - "Programmable ROM": memória programada por um dispositivo

programador de PROM. Programável uma única vez;

o EPROM - "Erasable PROM": memória que pode ser apagada e reprogramada

várias vezes. Apagável pela incidência de raios ultra-violeta e programável por um

dispositivo programador de EPROM;

o EEPROM - "Erasable Electrically PROM": memória EPROM eletricamente

modificável, sem necessidade de dispositivos externos apagadores ou programadores.

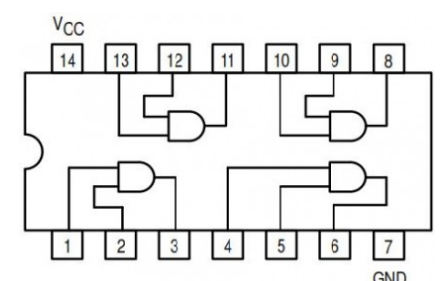
$$S1 = \bar{A} \cdot C' + B \cdot C$$

Situação	A	B	C	S
0	0	0	0	1
1	0	0	1	0
2	0	1	0	1
3	0	1	1	1
4	1	0	0	0
5	1	0	1	0
6	1	1	0	0
7	1	1	1	1

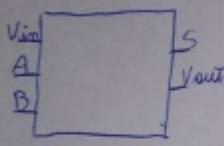
	\bar{B}	B
\bar{A}	1	1
A	0	1
	\bar{C}	C

$$S2 = C'D + B'D' + ABD$$

	\bar{C}	C
\bar{A}	1	1
A	1	1
	\bar{D}	D



03



Vin	A	B	S	Vout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$0+0+0=00$
 $0+0+1=01$
 $0+1+0=01$
 $0+1+1=10$
 $1+0+0=01$
 $1+0+1=10$
 $1+1+0=10$
 $1+1+1=11$

	B	B
A	0	1
A	1	0
	\bar{C}	C

$S = \bar{A}\bar{B}V_{in} + \bar{A}B\bar{V}_{in} + A\bar{B}V_{in} + ABV_{in}$
 $S = \bar{B}\bar{A}V_{in} + \bar{B}A\bar{V}_{in} + B\bar{A}V_{in} + BAV_{in}$
 $S = \bar{B}(\bar{A}V_{in} + A\bar{V}_{in}) + B(\bar{A}V_{in} + AV_{in})$
 $S = \bar{B}(A \oplus V_{in}) + B(A \oplus V_{in})$
 $Y = (A \oplus V_{in})$
 $S = \bar{B}Y + BY$
 $S = B \oplus Y$ $S = A \oplus B \oplus V_{in}$