

Um grupo de alunos da CDC é apaixonado por códigos binários e está desenvolvendo seu próprio código secreto para compartilharem mensagens entre si. Para isso, eles decidiram usar 5 bits para representar cada letra e número. Os códigos estão fazendo sucesso entre os amigos, mas, com o aumento do uso, alguns erros começaram a aparecer nas mensagens devido a bits digitados incorretamente. Isso gerou certa confusão, pois os amigos não conseguiam distinguir entre uma mensagem correta e uma mensagem com erro.

Preocupados com esses erros, eles resolveram incluir uma verificação adicional nas mensagens. Para garantir a integridade dos códigos, cada caractere agora inclui um **bit de paridade**, totalizando 6 bits por mensagem. Esse bit extra ajuda a detectar erros nos códigos, indicando quando uma entrada está incorreta.

Com essa ideia, eles decidiram pedir sua ajuda e de seu grupo para descrever um hardware usando **Verilog** que, além de exibir os caracteres em um display de 7 segmentos, também faça uma verificação de paridade. O circuito deve receber os 6 bits (5 bits do caractere e 1 bit de paridade) e, antes de mostrar o caractere no display, verificar se o código está correto. Se o código for válido, o display exibirá o caractere correspondente. No entanto, se houver um erro de paridade, o display mostrará um caractere especial para indicar que a mensagem contém um erro. Assim, os amigos poderão compartilhar suas mensagens em códigos binários com mais segurança, sabendo que erros nas entradas serão identificados imediatamente.

Os displays de sete segmentos oferecem uma maneira de baixo custo como alternativa a displays mais complexos e caros. Seus segmentos podem ser usados e combinados para criar formas simplificadas de algarismos arábicos.

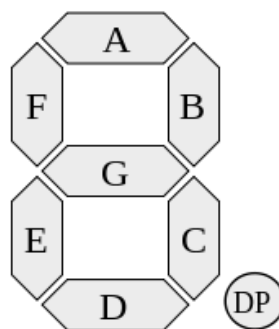


Figura 1: Display de sete segmentos.

No entanto, existe também um padrão de representação do alfabeto para tais displays, onde é possível de maneira simplificada representar cada uma das 26 letras, conforme ilustra a Figura 2.

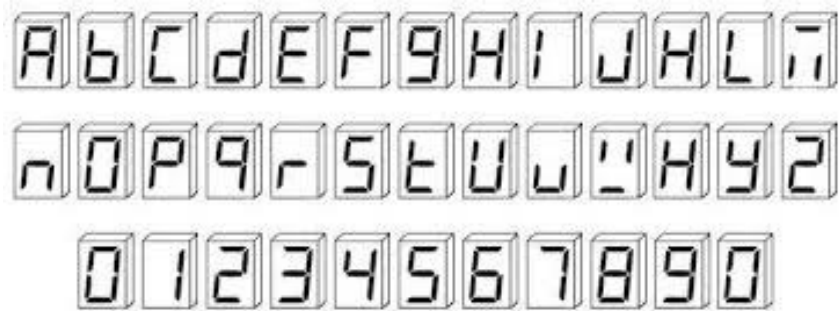


Figura 2: Definição dos caracteres.

Neste trabalho, você desenvolverá um circuito combinacional que inclui uma verificação de paridade para detectar possíveis erros na entrada de dados, além do mapeamento de caracteres para um display de 7 segmentos. A entrada total terá 6 bits, sendo 5 bits representando o caractere e 1 bit adicional de paridade para verificação de erro.

Para implementar o circuito de forma organizada, você deve dividi-lo em dois módulos: o **módulo de verificação de paridade** e o **módulo de mapeamento para o display de 7 segmentos**.

Módulo de Verificação de Paridade:

- Este módulo é responsável por verificar a paridade da entrada de 6 bits.
- Ele recebe os 6 bits como entrada, calcula a paridade dos 5 primeiros bits e compara com o bit de paridade fornecido.
- Produz uma saída de 1 bit que indica se a paridade é válida (1 para correta) ou se há um erro (0 para erro de paridade).

Módulo de Mapeamento para o Display de 7 Segmentos:

- Este módulo mapeia os 5 bits do código do caractere para os segmentos do display.
- Recebe os 5 bits do código como entrada e ativa os segmentos corretos no display de 7 segmentos para exibir o caractere correspondente.
- Também recebe o sinal de erro do módulo de verificação de paridade. Se o sinal de erro estiver ativo, exibe um caractere específico no display para indicar um erro de paridade.
- Se o código estiver fora do intervalo dos 20 caracteres válidos (códigos entre 10100 e 11111), o display deve permanecer apagado.

Comportamento do Circuito

1. **Entrada Válida (Paridade Correta e Dentro do Intervalo):** Se a entrada de 6 bits possui paridade correta e corresponde a um dos 20 códigos binários do grupo, o circuito exibe o caractere associado no display de 7 segmentos.
2. **Erro de Paridade:** Se a paridade da entrada de 6 bits não corresponder ao esperado (número ímpar de bits "1"), o circuito identifica um erro e exibe um caractere especial no display para indicar o erro. **O grupo deverá escolher uma caractere diferente dos 20 que são usados como válidos.**
3. **Entrada Fora do Intervalo (10100 até 11111):** Se o código de 5 bits está fora do intervalo dos 20 caracteres atribuídos e a paridade está correta, o módulo de mapeamento desativa todos os segmentos do display.

Para cada grupo será alocado uma ordem diferente de 20 caracteres aleatórios, incluindo números e letras, seguindo sempre a mesma ordem em bits, começando com 00000 e indo até 10011.

Tabela 1 - Subconjunto de caracteres a ser implementado por cada grupo

Bits	000 00	000 01	000 10	000 11	001 00	001 01	001 10	001 11	010 00	010 01	010 10	010 11	011 00	011 01	011 10	011 11	100 00	100 01	100 10	100 11
1	K	D	Y	O	3	F	W	R	7	A	L	Q	I	6	P	2	H	T	E	M
2	N	C	X	V	S	H	G	0	K	8	L	O	R	3	B	W	5	J	9	1
3	P	L	S	D	6	Q	T	A	Z	E	I	M	O	8	F	7	3	G	9	K
4	V	B	E	1	7	R	P	Y	0	K	M	H	S	Q	C	G	T	N	W	F
5	R	6	Q	P	5	D	Y	M	W	T	J	C	N	L	4	9	G	K	I	B
6	M	Y	V	H	T	0	A	S	Q	W	L	8	D	3	2	E	K	4	R	9
7	O	9	G	J	A	K	L	T	7	S	R	1	F	6	Q	B	M	C	W	V
8	T	F	L	I	P	W	2	Y	H	O	V	G	O	D	N	S	5	6	Q	R
9	S	A	4	M	G	V	O	P	K	1	J	9	X	F	7	W	L	3	C	T
10	J	Q	H	B	K	T	Y	I	F	8	W	6	C	3	N	9	7	0	R	O
11	C	S	8	L	R	P	Q	G	N	2	A	5	V	4	K	6	F	M	O	J
12	Y	O	3	T	M	N	X	9	P	H	K	J	A	L	W	S	2	7	Q	B
13	H	T	9	K	Q	I	V	F	W	C	4	B	1	M	G	P	N	D	0	R
14	E	W	7	0	6	Z	D	Q	M	F	X	I	B	8	T	L	A	S	P	O
15	N	F	I	W	Q	1	J	C	Y	G	H	L	V	T	K	P	3	A	O	4
16	5	G	X	A	T	P	M	V	L	O	C	D	E	8	W	B	Q	H	K	9
17	M	2	R	E	K	J	4	B	F	V	H	1	L	N	G	C	P	S	Q	T
18	P	A	D	F	X	8	0	M	H	S	L	V	N	9	5	O	3	Y	Q	G
19	G	L	K	N	Y	E	6	O	3	J	F	Q	H	T	9	D	M	W	S	2

20	V	R	B	1	M	9	E	T	D	6	O	I	K	4	F	8	X	Q	C	S
21	H	N	5	Z	G	M	L	S	V	W	B	R	F	D	P	I	7	3	C	K
22	T	3	L	O	9	Q	J	A	P	Y	F	X	B	0	R	6	G	V	K	W
23	F	Q	0	G	2	H	T	C	M	V	W	E	L	N	I	1	S	B	Y	K
24	S	B	4	Y	7	A	N	W	K	9	V	J	T	H	X	P	F	D	C	Q
25	D	6	K	P	L	Y	3	M	O	2	A	V	N	F	Q	W	1	T	R	E
26	R	M	F	8	C	0	L	7	W	E	K	I	O	P	3	T	9	X	Q	J
27	1	D	S	V	H	P	F	4	K	G	Q	N	O	Y	J	B	W	E	M	9
28	2	Y	P	O	X	R	6	D	V	S	K	F	L	Q	A	8	M	C	G	W

Entregas

1. Fase 1:

- Tabela verdade.
- Mapas de Karnaugh e expressões booleanas simplificadas.
- Diagramas de portas lógicas.
- Código em Verilog dos módulos implementados e do módulo de teste (testbench).
- Arquivo de simulação de ondas (.vcd).

2. Fase 2:

- Código para a implementação em FPGA.
- Documentação detalhada do processo e resultados obtidos.

Avaliação

A avaliação será feita com base nos seguintes critérios:

- 60% do total: Código fonte e implementação.
- 20% do total: Demonstração do funcionamento em FPGA.
- 20% do total: Documentação e apresentação.

Observações:

- É permitido que grupos comuniquem entre si sobre como resolver o trabalho, mas é **PROIBIDA** a cópia de quaisquer partes de código fonte e/ou documentação. Cópias serão exemplarmente punidas para quem copiou e quem forneceu o trabalho.

- Como não temos acesso a FPGA, a saída poderá ser mostrada através de 7 saídas binárias, cada uma representando um segmento sendo: 0 para um segmento apagado ou 1 para um segmento aceso.


Prazo de Entrega

- Fase 1: 26/11/2024
- Fase 2: 18/12/2024

Dúvidas

As dúvidas relacionadas deverão ser direcionadas ao Marcos (marcos.barbalho@ufv.br)

Tutorial de instalação das ferramentas necessárias para este e outros trabalhos da disciplina:

 Tutorial de Instalação das Ferramentas de ISL-OC