

Lab de Técnicas Digitais - Preparatório 3 - PUC-Rio

Rafael Rubim Cabral - 1511068

André Guimarães de Mello Alves - 1511032

1)

Resolvida direto no Circuit Maker.

2)

Resolvida direto no Circuit Maker.

3)

Nossa ideia inicial seguiu as seguintes etapas:

- 1) Somar dois números de 5 bits. O resultado terá 6 bits e será chamado de S.
- 2) Comparar S para ver se é menor que 19. Se positivo:
 - Ligar o LED CONV
 - Ligar um bit em algum lugar (o próprio CONV, por exemplo), onde será usado o fato " $S < 19$ "
- 3) Comparar S para ver se é maior que 9. Se positivo:
 - Ligar um bit em algum lugar, onde será usado o fato " $S > 9$ "
- 4) Agora teremos duas opções. Manter o resultado S em 6 bits binários ou transformá-lo em um número de 8 bits em BCD. Decidimos usar sempre 8 bits de saída no mesmo display: se a saída não estiver em BCD, apenas 6 desses 8 bits representarão a soma (os 2 mais significativos estarão zerados).
- 5) Em todo caso, somaremos $S + 6$. Usaremos esse resultado caso S precise ser transformado em BCD.
- 6) Por fim, utilizar um multiplexador para escolher uma entre as duas possibilidades: "S" ou "S+6". Para saber qual delas escolher, podemos testar os 2 fatos: se " $S < 19$ " e " $S > 9$ ", então precisamos transformar S em BCD (no caso $S \leq 9$, S já está em BCD). Caso contrário, não há necessidade de fazer transformação alguma.

Essa foi a solução inicial, porém escolhemos algumas otimizações:

- Ao invés de usar um multiplexador para escolher entre "S" e "S+6", decidimos usar a saída como "S+X". Usaremos um somador que soma X = 6 caso queiramos transformar S em BCD ou soma X = 0 caso S já esteja em BCD ou não queiramos transformar S em BCD.
- Ao invés de utilizar dois comparadores para testar os fatos "S < 19" e "S > 9", preferimos montar um subcircuito de entrada S cuja saída é 1 caso S esteja no intervalo desejado (9 < S < 19) e 0 caso contrário. Se a saída do subcircuito for 1, então S deve ser transformado em BCD. Caso contrário, não. Para montar esse subcircuito, fizemos o seguinte teste: se o bit mais significativo de S (chamado CARRY) estiver ligado, então S > 19, logo o resultado do subcircuito é 0. Com os outros 5 bits menos significativos de S fizemos um mapa de Karnaugh para testar (9 < S < 19). Vamos chamar o resultado do circuito mínimo em 2 níveis desse mapa de Karnaugh de Y = 1 se S estiver no intervalo desejado e Y = 0 se S estiver fora do intervalo. Então o resultado do subcircuito será 1 se (CARRY = 0 AND Y = 1) e 0 caso contrário. Chegamos à conclusão, experimentalmente, de que montar esse subcircuito gastará menos chips na prática do que fazer 2 comparações com números de 6 bits. Para gastar ainda menos chips na prática, substituímos o (CARRY = 0 AND Y = 1) por $\sim(\text{CARRY} = 1 \text{ OR } Y = 0)$. Assim, podemos aproveitar as entradas do chip OR que estão sobrando.

Segue o resultado do mapa de Karnaugh de 5 entradas que utilizamos para implementar o subcircuito:

| | \overline{DE} | \overline{DE} | DE | DE |
|------------------|-----------------|-----------------|------|------|
| \overline{ABC} | 0 | 0 | 0 | 0 |
| \overline{ABC} | 0 | 0 | 0 | 0 |
| \overline{ABC} | 1 | 1 | 1 | 1 |
| \overline{ABC} | 0 | 0 | 1 | 1 |
| \overline{ABC} | 1 | 1 | 0 | 1 |
| \overline{ABC} | 0 | 0 | 0 | 0 |
| ABC | 0 | 0 | 0 | 0 |
| ABC | 0 | 0 | 0 | 0 |

| Grupos | |
|---------------|------------------------------|
| (10,11,14,15) | \overline{ABD} |
| (12,13,14,15) | \overline{ABC} |
| (16,17) | $\overline{ABC}\overline{D}$ |
| (16,18) | \overline{ABCE} |

Função resultado:

$$Y = \overline{ABD} + \overline{ABC} + \overline{ABC}\overline{D} + \overline{ABCE}$$

Com essa função, pode-se implementar os gates diretamente no modelo.