

Atividade – Pacotes e Componentes – 23/04/2019

Para os exercícios abaixo relacionados, o arquivo contendo a top level entity deverá conter somente sinais e mapeamentos, ou seja, todos os circuitos deverão estar na forma mapeada, em componentes. Da mesma forma, o número de bits deverá estar na forma generic para prover flexibilidade. **Avaliação: 05/05/2019 em equipes de no máximo três participantes.**

1. Desenvolver e implementar na Basys 2 um conversor binário \rightarrow *gray* de quatro *bits*. Utilizar como entrada as chaves SW4 (MSB) a SW1 (LSB) nesta ordem. Utilizar como saída os *leds* LD4 (MSB) a LD1 (LSB) nesta ordem.
2. Desenvolver e implementar na Basys 2 um decodificador hexadecimal / 4 bits \rightarrow 7 segmentos usando pacotes e componentes. Utilizar as mesmas entradas do Exercício 1 e o *display* de sete segmentos #3 como saída.
3. **APS:** Desenvolver e implementar na Basys 2 um decodificador BCD ou hexadecimal, ambos de 4 bits \rightarrow 7 segmentos usando pacotes e componentes.
 - a. Utilizar a chave SW5 para alternar a decodificação entre hexadecimal e BCD;
 - b. Durante os momentos em que o decodificador estiver operando com entrada em BCD, apresentar “E” como código especial para os casos em que alguém inadvertidamente entre com uma combinação de valores nas chaves que esteja acima de “1001”;
 - c. As chaves SW7 e SW6 serão responsáveis por selecionar qual *display* de 7 segmentos exibirá a saída decodificada;
 - d. As chaves SW4 a SW1 serão usadas como a entrada de dados, seja ela em hexadecimal ou em BCD;
4. **ATV:** desenvolver e implementar na Basys 2 os seguintes circuitos:
 - a. Multiplicador de dois ou três bits com resultados no *display* de 7 segmentos
 - b. Divisor de dois ou três bits (quociente no *display* mais à esquerda e resto no *display* mais à direita)
 - c. Circuito y^x de dois bits com resultado nos *leds* SW7 ao SW1
 - d. Circuito $y \ll x$ (deslocamento à esquerda) de dois bits com resultado nos *leds* SW5 ao SW1