

LÓGICA DIGITAL (1001351)

EXPERIMENTO NR.6

Somador/Subtrator, Decodificador e Comparador/Deslocador ¹

Aviso ENPE

Esta prática foi adaptada para ser realizada em simuladores. As instruções originais foram mantidas para que se tenha uma ideia do procedimento real. Deve-se destacar quando não foi possível obter a informação solicitada por limitação do simulador usado.

1 Instruções Gerais

- Grupos definidos no AVA, só incluir os nomes de quem efetivamente participou;
- Ler atentamente todo o procedimento desta experiência antes de realizá-la;
- Construir um desenho esquemático completo do projeto, incluindo nomes para os blocos e ligações, antes de codificar;
- Template e testes disponíveis no [EDApayloadground](#).

2 Objetivos da Prática

- Implementação de somador/subtrator de 4 bits, decodificador e comparador/deslocador utilizando como saída [o display de 7 segmentos²] o valor em hexadecimal e também decodificado.
- Realizar o desenho (não precisa entregar) e a implementação hierarquia de módulos.
- Simulação do projeto realizado através de um arquivo de *test bench* e análise de sua resposta temporal.
- Criar casos de testes adicionais conforme especificação adiante.
- **Implementação do projeto no Kit de desenvolvimento Zybo Z7-20.²**

¹Revisão 19/09/2021: Prof. Ricardo Menotti, Prof. Maurício Figueiredo

²Opcional, usando o laboratório remoto

3 Procedimentos Experimentais

Deseja-se implementar um Somador/Subtrator de 4 bits, um Decodificador e um Comparador/Deslocador, utilizando como saída o valor em Hexadecimal e Decodificado (ver exemplo no values.tv). O módulo deve ter as seguintes características:

- Entradas:
 - A (4 bits com sinal)
 - B (4 bits com sinal)
 - Op (1 bit / 0=somar / 1=subtrair)
- Saídas:
 - S (soma/subtração entre A e B com 4 bits com sinal)
 - *Overflow* (1 bit / somente da soma/subtração)
 - D (16 bits / valor da soma S decodificado)
 - T (4 bits com sinal), onde:
 - * Se $A > B$, então $T = A \gg 1$
 - * Senão $T = B \ll 2$
- Observações:
 - O *overflow* só é sinalizado para a soma/subtração, não sendo influenciado pelos deslocamentos (quando perdem-se bits);
 - O arquivo de testes (values.tv) está em hexadecimal e os valores são interpretados com sinal, portanto:
 - * Números acima de 7 são negativos (intervalo -8 a 7);
 - * Para mostrar o deslocamento em binário na porta D, pegamos 1 a cada 4 bits do arquivo que está em hexadecimal³.

³Lembrar da diferença entre `readmemh` e `readmemb`.

Inclua mais 10 casos de teste distintos no arquivo de testes, que contenham somas, subtrações, com e sem *overflow*, com deslocamentos para os dois lados de números positivos e negativos, com a seguinte organização de experimentos:

- com operações de adição (sem *overflow*), considerar pares de parcelas com os seguintes sinais (positivo/negativo): (+,+) (-,-) (+,-)
- com operações de subtração (sem *overflow*), considerar pares minuendo/subtraendo com os seguintes sinais (positivo/negativo): (-,-) (-,+) (+,-)
- com operações de adição e *overflow*, com os seguintes exemplos em termos de sinais (positivo/negativo): (+,+) (-,-)
- com operações de subtração e *overflow*, com os seguintes exemplos em termos de sinais (positivo/negativo): (+,-) (-,+)

A saída do *test bench* na console deve ser em forma de tabela, apresentando os casos de teste, com a devida resposta e verificação de erros. Envie o link com o projeto.