Projeto 1- PSI 3451

Use o material de apoio do arquivo PDF, **Implementando_LFSRs _Galois_20.**

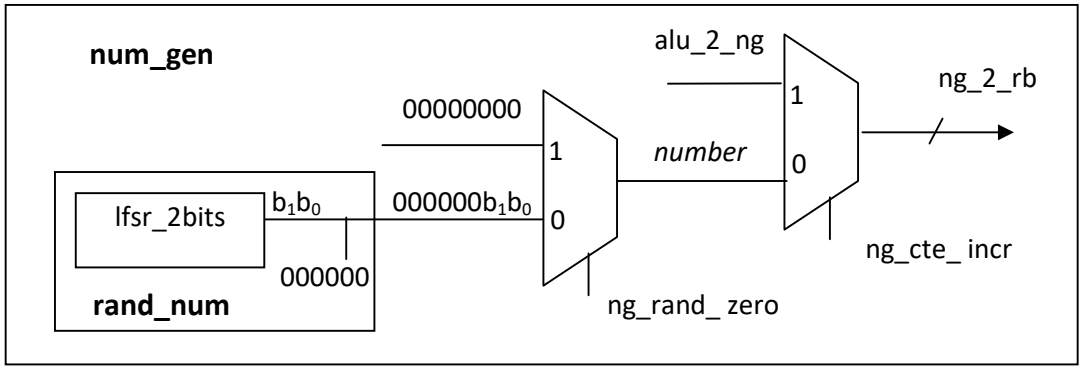
Data de Entrega de Relatório: **a ser definida**

Módulo Random_num

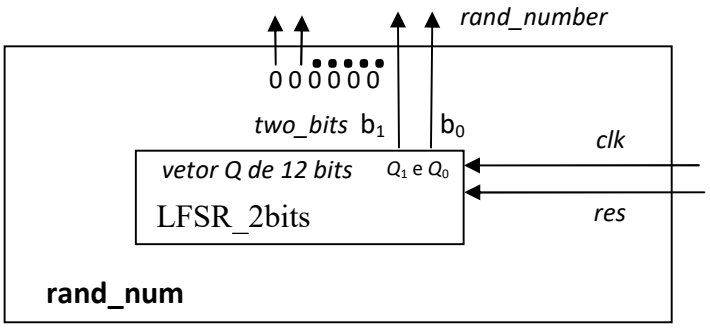
O Módulo VHDL *Random_num* a ser projetado é um gerador de números aleatórios que deverá ser integrado ao Circuito do Wisdom, substituindo a função de geração de números aleatórios testada junto ao bloco *Num_Gen* (aula 4).

A figura 1 ilustra as interfaces do módulo *Rand_num* e a sua esperada conexão dentro do *Num_Gen* e do datapath do Base_circuit do Wisdom. A figura 1.a mostra o esquema utilizado na Aula 4 (copiado da Figura 1 da apostila de conceitos), com dois bits do LFSR sendo concatenados a zeros dentro do bloco *Rand_num*.

Dos n(=12) bits do LFSR, gerador de números aleatórios a ser projetado, apenas 2 serão utilizados para a definição do endereço inicial de memória do Guru, na forma indicada pela figura 1.b. Ela indica também que os bits 0 e 1 (Q_0 e Q_1) do LFSR devem ser selecionados; o fato é que qualquer dois bits do LFSR poderiam ser selecionados, tendo efeito similar. Esta seleção de bits traz uma consequência interessante do ponto de vista de aleatoriedade. Enquanto a sequência de números nos 12 bits do LFSR é pseudoaleatória, i. e., consiste de um conjunto de vetores distintos que vai se repetindo, o uso de um número menor de bits tem a aparência de ser aleatório devido ao efeito de *aliasing*. No caso de 2 bits, é bem possível que dois ou mais vetores consecutivos apresentem o mesmo valor, o que não seria possível com os 12 bits do LFSR.



(a) Esquema de geração de número aleatório da aula 4



(b) Esquema de geração de número aleatório com LFSR

Figura 1

O reset (res) do circuito deve ser utilizado para forçar a saída dos FFs em "1" (equivalente a um *set* do FF), na forma da Figura 2. Ou seja, o LFSR da configuração Galois deve ser adaptado para impor o seed de "111...111".

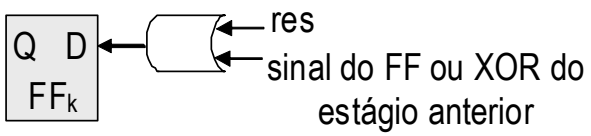


Figura 2

O trabalho consiste de 6 etapas.

1. Calcular os coeficientes do polinômio gerador do seu LFSR.
2. Simular em software o LFSR de forma a compreender e conhecer a sua sequência pseudoaleatória.
3. Escrever o código VHDL estrutural do bloco *rand_num* da Figura 1. Deve conter um *linear feedback shift register* (LFSR) genérico, personalizando o código para o seu polinômio utilizando **obrigatoriamente** o comando GENERATE. A saída do bloco *rand_num* deverá ser um vetor binário de 8 bits concatenando o vetor '000000" com os dois LSBs do LFSR.
4. Demonstrar que a sequência obtida é a correta por simulação (testbench).
5. Preparar um relatório descritivo das etapas e conclusões. Os alunos deverão seguir um modelo de relatório a ser fornecido, que deverá ser preenchido com os resultados listados a seguir.

Importante: Cada aluno deverá personalizar o LFSR de acordo com o seu #USP.

Abaixo estão as características do projeto.

a) O LFSR de 12 FFs será do tipo Galois, como descrito no material adicional, correspondendo a um polinômio de grau 12, da forma $(x^{12}+a_{11}.x^{11}+....a_1.x^1+1)$. Os coeficientes $a_{11}...a_1$, serão definidos de acordo com o #USP de cada aluno. Lembrando que os coeficientes a_{12} e a_0 são sempre =1.

b) Cálculo dos coeficientes do polinômio:

- Calcule X= #USP mod 2048 (resto da divisão #USP/2048).
- Sendo X um número decimal converta-o em número binário de **11 *bits***. O dígito binário mais significativo corresponderá ao coeficiente c_{11} e o menos significativo ao coeficiente c_1.
- Caso X=0, adote os 3 dígitos menos significativos do seu #USP como seu valor de X.

Vamos dar dois exemplos.

- Exemplo 1:
  - $X_{decimal}= 53 \rightarrow X_{binario}= 00000110101$ ($a_{11}=0$ e $a_1=1$)
    que corresponde ao polinômio:
    $$x^{12}+x^6+x^5+x^3+x^1+1$$
    em que foram acrescentados o $a_{12}=1$ e o $a_0=1$

- Exemplo 2:
  - $X_{decimal}$=2044 → $X_{binario}$= 11111111100 (a_{11}=1 e a_1=0) que corresponde ao polinômio:
    $x^{12}+x^{11}+x^{10}+x^9+x^8+x^7+ x^6+x^5+x^4+x^3+1$
- Exemplo 3 (resto 0):
  - Número USP= 9584640→ $X_{decimal}$= 0 → obter novo X
  - $X_{decimal}$= 640 → $X_{binario}$= 01010000000 (a_{11}=0 e a_1=0) que corresponde ao polinômio:
    $x^{12}+x^{10}+x^8+1$

Relatório do projeto: o aluno deve apresentar o polinômio, explicando como chegou à configuração (atenção: revise e confirme que o polinômio obtido é o correto; em caso de erro, haverá comprometimento dos demais resultados e, portanto da nota).

c) Conhecido o seu polinômio, deve-se estruturar o circuito LFSR composto de DFFs e XORs. Faça uma pequena simulação manual do LFSR, de 2 a 3 ciclos, com semente= 11...111, para determinar a sequência de valores.

Relatório do projeto: apresentar um esboço (em forma digital ou manuscrito) do esquema do circuito LFSR desenvolvido.

d) Gere a sequência de bits gerados através do software on-line descrito no documento de descrição do LFSR Galois. Capture as imagens do software com os resultados dos **10 primeiros ciclos** após a inicialização (em '1111...111') para serem apresentados. Compare os resultados com os poucos simulados no item c) para ter certeza que está tudo certo.

Relatório do projeto: impressão das imagens de tela com os resultados da simulação (10 ciclos). Reportar em representação hexadecimal.

e) Faça a descrição equivalente do circuito *rand_num* em VHDL (figura 1.b) seguindo o esquema dado no item c) (atenção: utilize os mesmos nomes no diagrama e no VHDL, para os sinais e portos).

- O DFF é fornecido ao aluno (na área da disciplina no Moodle).

- Para o XOR e OR, copie os módulos utilizados com os somadores das aulas anteriores. Reaproveite os códigos DEVIDAMENTE ADAPTADOS para o projeto (número de portos e seus nomes).

- Você deverá usar **obrigatoriamente** o comando **GENERATE** para construir o conjunto do LFSR. O comando deverá ser usado de forma a otimizar a codificação (**não usar o GENERATE na forma trivial**).

- O vetor de estados conterá os bits de saída dos FFs (Q11 a Q0, seguindo o documento de descrição do LFSR Galois) e portos do módulo deverão ser de acordo com o especificado na figura 1.b.

Relatório do projeto: incluir a descrição do projeto em VHDL.

f) Prepare o testbench para validar o seu código VHDL (o módulo rand_num e os bits do LFSR) através do simulador ModelSim. Um arquivo topo será fornecido aos alunos que deverão codificar o arquivo de estímulos. Gere cartas de tempo no **Wave** de forma a mostrar uma sequência de estados (com o vetor de estados). Simule pelo menos por vinte (20) ciclos de relógio de evolução da sequência, após a ativação do sinal set.

Relatório do projeto:

1) incluir a descrição VHDL dos arquivos de *testbench* com seus componentes. O código VHDL deve conter comentários que explicitem as intenções do autor nas diversas partes, i.e, devem permitir entender o funcionamento do *testbench*.
2) impressão **legível**[1] da carta de tempos com pelo menos 20 ciclos de relógio a partir da inicialização (deixe os sinais importantes evidentes no Wave). Apresente tanto os sinais das saídas **Q** dos FFs (12 bits do LFSR) como de saída ***rand_number*** do módulo *rand_num* (de 8 bits).

   Obs. Represente no **Wave** os números aleatórios gerados com a mesma base numérica utilizada no item **d)**, para possibilitar a comparação.

g) Validação por comparação entre o resultado de seu projeto e o do software.

Relatório do projeto: Discuta e demonstre que os resultados da simulação do projeto e da execução do software rodado são os mesmos (compatíveis entre si).

h) Comparação entre aleatoriedades.

Relatório do projeto: Liste os 20 resultados da simulação do LFSR em f) para as saídas **Q** e ***out***. Analise as ocorrências e/ou possibilidade de *aliasing*, ou seja a diferença entre as aleatoriedades dos vetores ***Q*** e ***rand_number***.

[1] Cartas de tempo não legíveis ou sem indicações claras serão consideradas como não entregues.