LFSRs - configuração Galois (PSI-3451/2018)

Um linear feedback shift-register, LFSR, é um registrador de deslocamento que possui realimentação entre as saídas de um conjunto de elementos de armazenamento e as suas entradas, a partir de realização de operações lógicas. LFSRs são usados de forma generalizada para geração aleatória de vetores. Existem dois tipos de LFSRs, de acordo com a configuração da realimentação utilizada, conhecidos na literatura por configurações Fibonacci e Galois. Neste texto trataremos apenas da configuração Galois, ficando por conta do aluno interessado uma pesquisa mais detalhada sobre o assunto, para o qual existe farta documentação na Internet e literatura técnica especializada.

1. Geração de LFSR e Polinômio Primitivo

Na configuração Galois, o LFSR é um registrador de deslocamento onde o próximo estado de todos os elementos de armazenamento, da direita para a esquerda (LSB para o MSB), como mostrado na Figura 1, é igual a

$$D_i=Q_{i-1}\oplus a_iQ_{n-1}, \text{ para } i = 1..n-1 \qquad \text{(Equação 1)},$$

uma combinação entre o estado atual do elemento de armazenamento na posição anterior da cadeia e do estado atual do último elemento da cadeia. Excetua-se o primeiro registrador (da esquerda) que recebe o valor deslocado do mais à direita diretamente. Caso uma ligação a_i estiver inativa, ou seja, $a_i=0$, o estado futuro será o valor do estado atual do elemento de armazenamento na posição anterior da cadeia, ou seja,

$$D_i=Q_{i-1}, \text{ para } i = 1..n-1 \qquad \text{(Equação 2)}.$$

Se a ligação existir, com $a_i=1$,

$$D_i=Q_{i-1}\oplus Q_{n-1}, \text{ para } i = 1..n-1 \qquad \text{(Equação 3)}.$$

Em um determinado estado, caso o valor do MSB seja $D_{n-1}=0$, indicando que todos os operadores XOR terão a entrada da realimentação com valor lógico '0', novamente,

$$D_i=Q_{i-1}, \text{ para } i = 1..n-1 \qquad \text{(Equação 4)}.$$

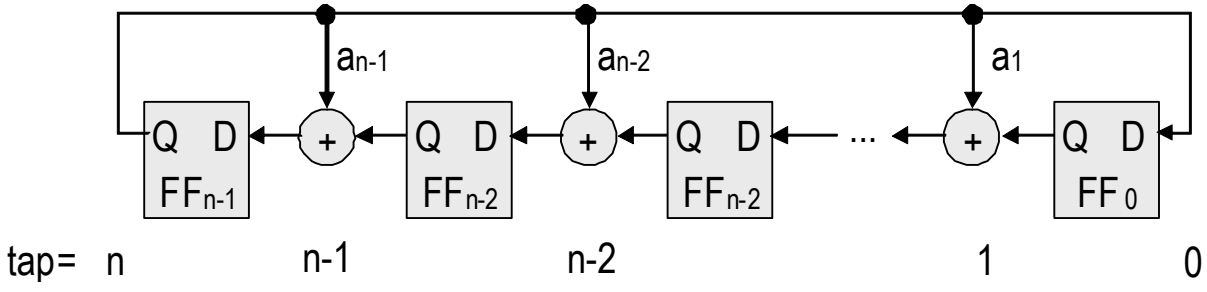

Figura 1. LFSR- configuração Galois

Cada uma das ligações existentes é conhecida por *tap* que recebe a enumeração de acordo com a sua posição. Um LFSR pode ser identificado/definido pelos seus *taps*, apesar de que não há um padrão exato de como se expressar isto. Adota-se aqui um padrão compatível

com o software online a ser apresentado mais adiante. O exemplo da Figura 2 ilustra o caso de LFSR (4, 3, 2). Observe-se que o MSB, no caso 4, sempre aparece na enumeração de *taps*, apesar de não haver um XOR nesta posição; ela é necessária para estabelecer o tamanho do LFSR.

A equação de estados do LFSR pode ser expressa por um polinômio característico na forma

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + a_{n-2} x^{n-2} + \ldots + a_1 x + a_0$$

onde $a_0=1$ e $a_n=1$ para um polinômio de grau *n* (ou LFSR de grau *n*). Os demais coeficientes expressam a existência ou não das realimentações. Então,

$$p(x) = x^n + a_{n-1} x^{n-1} + a_{n-2} x^{n-2} + \ldots + a_1 x + 1$$

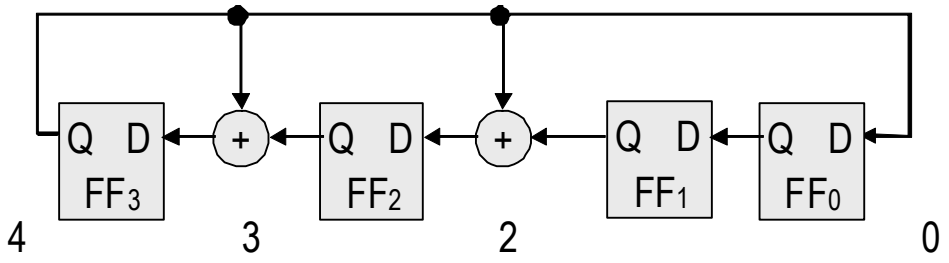

Figura 2. LFSR (4, 3, 2)

Por simplicidade, usamos como coeficientes os mesmos parâmetros indicativos de ligação da figura 1. Desta forma, o polinômio característico do LFSR da figura 2 é

$$p(x) = x^4 + x^3 + x^2 + 1$$

A máquina de estados gera uma sequência pseudo-aleatória de vetores representados pelos bits das saídas dos registradores (estado atual), Q_{n-1}, Q_{n-2},...Q_1, Q_0. Há duas questões bastante relevantes no uso dos LFSRs como geradores de sequências aleatórias:

1) o tamanho da sequência: a teoria diz que a sequência aleatória é máxima somente quando o polinômio é primitivo, ou seja, ele não é divisível por outro polinômio qualquer. Por exemplo, o polinômio de LFSR (4, 3) é primitivo, com uma sequência de tamanho n^4-1=15. Quando o polinômio não é primitivo, pode-se ter várias sequências independentes de tamanhos menores. Um padrão de zeros (0, 0.., 0) manter-se-á imutável, ciclo a ciclo, correspondendo a uma sequência de tamanho 1) (esta é a razão da sequência máxima ser de apenas 15 vetores).

2) a semente: os LFSRs devem adotar uma condição ou estado inicial para os registradores. Como visto no item anterior, se a semente for (0,0,..,0), não ocorrerá nenhuma sequência útil; o estado inicial (1,1,..1,1, por exemplo) faz com que alguma sequência alternativa ocorra (de tamanho máxima, se o polinômio for primitivo).

2. Simulação por software on-line (https://leventozturk.com/engineering/crc/**)**

O sítio da Internet acima apresenta opções de cálculo on-line de geradores aleatórios, assim como a obtenção automática de código de hardware e software em alguma linguagens de referência. Inclui-se aí o gerador baseado no LFSR-Galois. Este recurso será utilizado pelo aluno durante o teste de hardware projetado, para aumentar a confiabilidade do projeto implementado.

A utilização dos recursos on-line é razoavelmente autoexplicativa, porém, para maior produtividade no seu uso, alguma dicas são adiantadas a seguir:

1) Bloco Configure:

- Atentar que há necessidade de se selecionar "Galois LFSR" no campo *Type* a cada computação. Infelizmente, o programa não guarda a último tipo utilizado e sempre retorna à opção CRC.

- Para o polinômio, utilize o formato X_n to X_0. Todos os n taps devem estar presentes: os de índice n e 0 são obrigatórios.

- Para a *seed* (semente) no campo *Initialise*, o bit mais à esquerda é o MSB, (Q_{n-1} do LFRS) e o mais à direita é o LSB (Q_0 do LFRS). É obrigatória a adição de mais um "0" à esquerda da semente (para se ter n bits para a computação), apesar de ele não ter papel real no LFSR. Simplesmente, desconheça-o quando for interpretar o movimento da semente.

- *Data Width* deve ser um valor entre "1" e "63". O simulador on-line realiza a computação de uma sequência de tamanho por Data Width, porém só o último padrão é mostrado. Para verificar a sequência toda a partir da semente, deve-se repetir a computação para valores crescentes de Data Width. (Obs. Deixe o campo *Process Direction* no default d[n] to d[0]; parece não ter efeito nos cálculos para um LFSR Galois)

2) Bloco Generate Code - não é usado

3) Bloco Calculate Output:

- Campo *Input Data* : não se aplica a LFSR Galois

- Para o campo *Output Format*, adote o[n] para o[0], em razão de manter o padrão [MSB,..,LSB] do *tap* e *seed* (semente)

- O campo *Galois LFSR output* apresenta o padrão final após a sequência de padrões de tamanho definido por *Data Width*. Estará no formato o[n] para o[0], como selecionado acima.