INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA - CAMPUS SÃO JOSÉ

CURSO ENGENHARIA DE TELECOMUNICAÇÕES

DISCIPLINA: Sistemas de comunicação I - COM029007

AP2 - Modificações Timer

Alunos: Jefferson Botitano Calderon R e Leonardo Ludvig Da silva

Professor: Mario de Noronha Neto

São José, 2022

O projeto timer

O projeto TIMER foi apresentado como tendo um contador de minutos e segundos em binário com seus registradores para armazenamento de informação e troca de estado do timer. Sendo assim composto por outros componentes como bcd2ssd que converte um número em formato BCD (Binary Coded Decimal) para um número em formato SSD (Seven-Segment Display), um componente bin2bcd que recebe um número em formato binário como entrada e produz um número em formato BCD como saída. O componente pode ser implementado de várias maneiras, usando operações lógicas para converter cada dígito binário em um grupo de 4 bits binários correspondentes.

Adicionando milésimo de segundo

Para esta etapa foi necessário adicionar o componente milésimo de segundo ao componente timer, como o projeto está utilizando incrementadores em binário foi mantido a lógica já feita de timer para segundo e minuto,porém para o caso do milésimo de segundo é necessário que a contagem seja até 499 e em seguida passe para o próximo estado ao chegar no seu limite. Outra alteração foi feita no port map do timer sendo requerido adequá-lo para o milésimo de segundos, juntamente foi feita a adição de componentes hexadecimais para suporte dos milésimos de segundos e a necessidade de adição de três componentes juntamente com seu conversor bin2bcd, sendo assim os milésimos de segundos que possuem saída binária será convertida para BCD que será projetado no BCD2SSD, a Figura 1 demonstra a simulação para o funcionamento dos componentes.

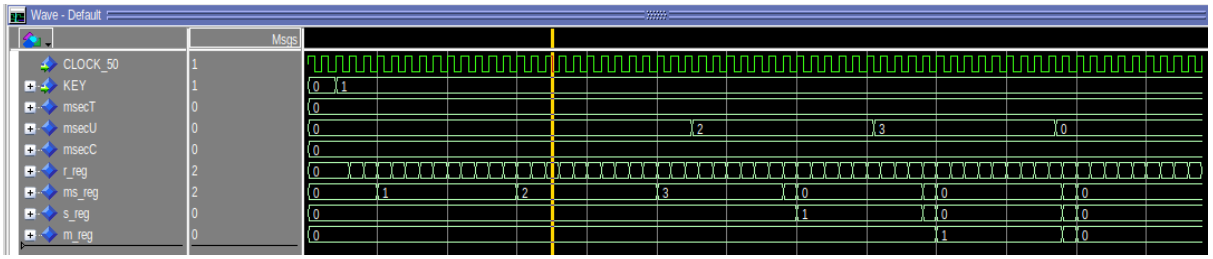

Figura 1.

Adicionando arquivo PLL

A etapa de adição de um arquivo PLL que tem o intuito de sincronizar um sinal de clock de entrada com um sinal de referência, produzindo um sinal de clock de saída com uma frequência e fase precisamente controladas. O PLL é amplamente utilizado em projetos de eletrônica digital para garantir que os sinais de clock necessários para o funcionamento do sistema sejam gerados de forma confiável. Sendo assim a estrutura dos componentes foi mantida e apenas este componente foi adicionado, a figura 2 demonstra a nova simulação a partir do novo sinal de clock gerado pelo PLL.

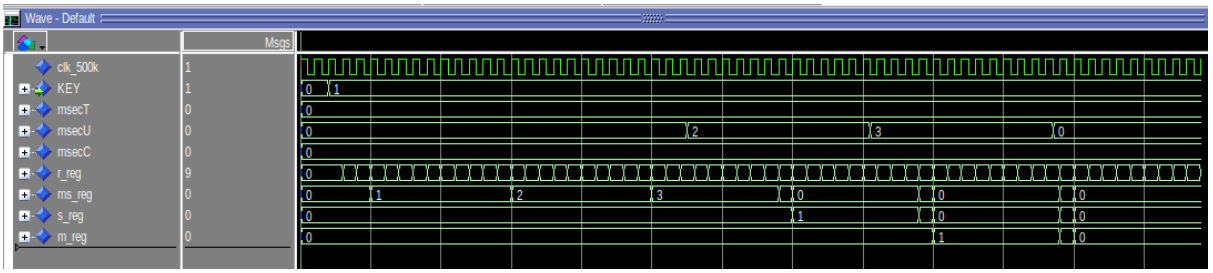

Figura 2.

Alterando contadores do projeto para BCD

Nesta etapa do projeto foi realizado a reestruturação de utilização de binário para BCD, sendo assim o componente bin2bcd não será mais necessário, para isso foi necessário realizar as condicionais aritméticas dos contadores. A figura 3 representa a simulação feita pela nova estruturação dos contadores de milissegundo tendo unidade,dezena e centena e para segundo e minuto apenas unidade e dezena.

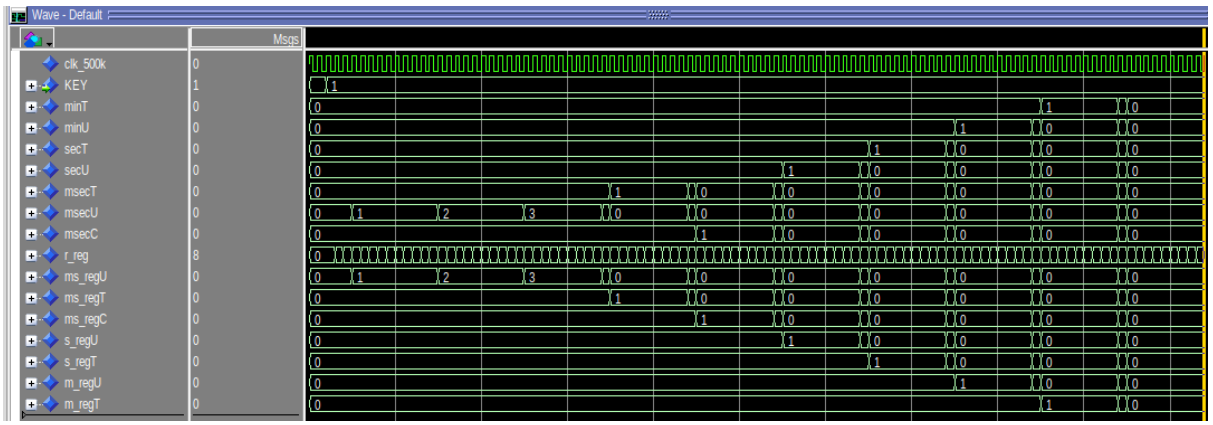

Figura 3.

Alterando registrador r_reg para LFSR

Ao realizar a modificação do nosso registrador r_reg para um LFSR, foi necessário primeiro identificar quantos bits desejamos para identificar os TAPS da nossa lógica de mudança de estado utilizando o novo registrador, para isso foi necessário consultar a tabela da figura 4.

# of Bits	Length of Loop		Taps
2	3	*	[0,1]
3	7	*	[0,2]
4	15		[0,3]
5	31	*	[1,4]
6	63		[0,5]
7	127	*	[0,6]
8	255		[1,2,3,7]
9	511		[3,8]
10	1,023		[2,9]
11	2,047		[1,10]
12	4,095		[0,3,5,11]
13	8,191	*	[0,2,3,12]
14	16,383		[0,2,4,13]
15	32,767		[0,14]
16	65,535		[1,2,4,15]
17	131,071	*	[2,16]
18	262,143		[6,17]
19	524,287	*	[0,1,4,18]
20	1,048,575		[2,19]
21	2,097,151		[1,20]
22	4,194,303		[0,21]
23	8,388,607		[4,22]
24	16,777,215		[0,2,3,23]
25	33,554,431		[2,24]
26	67,108,863		[0,1,5,25]
27	134,217,727		[0,1,4,26]
28	268,435,455		[2,27]
29	536,870,911		[1,28]
30	1,073,741,823		[0,3,5,29]
31	2,147,483,647	*	[2,30]
32	4,294,967,295		[1,5,6,31]

Figura 4.

Sendo assim ao estarmos utilizando 9 bits, descobrimos que é necessário utilizar os TAPS 3 e 8, para que com isso possamos identificar as posições de bits na figura 5 por meio da tabela gerada de TAPS e número de bits utilizados que as posições desejadas são dos bits 5 e 0.

TAPs	0	1	2	**3**	4	5	6	7	**8**
BITs	8	7	6	**5**	4	3	2	1	**0**

Figura 5.

Em seguida necessitamos descobrir o ponto de partida do antigo registrador com o novo registrador LFSR, por meio da simulação da figura 6 é possível identificar que quando o registrador r_reg atinge 499 o registrador LFSR está em 140, sendo assim ele será nossa começo de partida para poder prosseguir de forma adequada com os contadores já existentens.

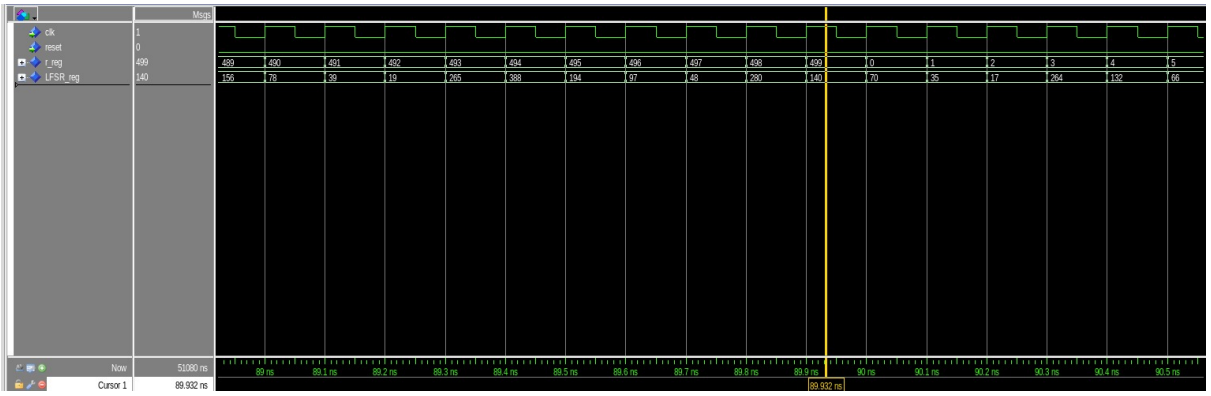

Figura 6.

A figura 7 é o resultado final da substituição do nosso antigo r_reg pelo novo LFSR, é possível identificar o mesmo comportamento porém em níveis de componentes temos uma tremenda diferença.

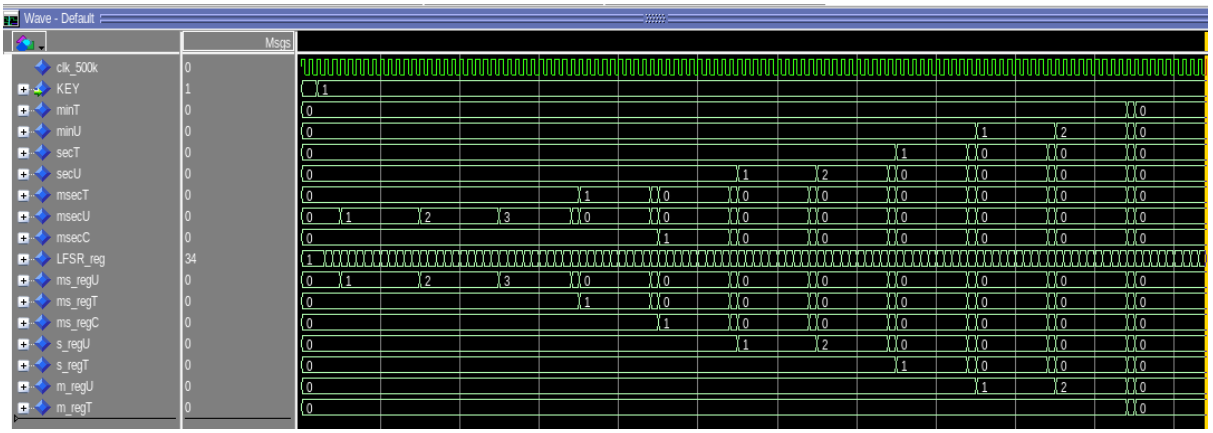

Figura 7.

Conclusão

Ao decorrer das modificações feitas é notório que o número de registradores foi diminuindo juntamente com seus elementos lógicos, na parte 1 temos a estruturação de partida do projeto, ou seja , temos como deve ser o comportamento desejado de nossa aplicação e as demais em seguidas são suas otimizações. A parte 2 do projeto consiste em diminuir o número de elementos lógicos e registradores por meio da adição do componente PLL para clock, ou seja, sem um PLL seria necessário utilizar vários componentes como osciladores, divisores de frequência e circuitos de sincronização de fase que para gerar os sinais de clock de alta qualidade e precisão necessários para o sistema. Cada um desses componentes adiciona complexidade e demanda mais recursos do sistema, incluindo espaço, energia e tempo de projeto.

Em relação a otimização da parte 3 temos como a substituição da etapa de conversão dos binários e como estão em BCD é mais rápido os processos aritméticos, poupando assim número de registradores para armazenamento dessas informações durante o processo. A parte 4 consiste na adição do registrador LFSR que utiliza um processo de deslocamento com feedback para gerar uma sequência pseudoaleatória de números binários. A saída do registrador é obtida a partir de uma combinação lógica dos bits armazenados em diferentes posições do registrador, de acordo com uma configuração pré-definida. A configuração do LFSR pode ser escolhida de forma a garantir que a sequência gerada tenha propriedades estatísticas desejáveis, como uma distribuição uniforme dos valores, sendo que no nosso caso foi feita a partir do r_reg anterior como ponto de partida, juntamente com isso a frequência pode aumentar a frequência do sistema, pois o processo de geração de números pseudoaleatórios pelo LFSR pode ser executado de forma mais rápida e eficiente do que o processo convencional de geração de números aleatórios. Isso pode permitir que o sistema opere em uma frequência mais alta, reduzindo a latência e aumentando a taxa de transferência de dados, melhorando a eficiência do sistema. A tabela 1 representa exatamente o que foi descrito nesta conclusão por meio das notas feitas e registrados de elementos lógicos, registradores e frequência.

	Binário	**Binário**	**BCD**	**BCD**
CLK	50MHz	500KHz	500KHz	500KHz
LE	285	258	124	120
Register	129	123	51	37
Freq (MHZ)	155.67	100.42	47.94	73.17

Tabela 1.