

Leonardo Mendonça de Araujo,
Lucas Bagatini do Nascimento,
Mário Muramatsu Júnior

RELATÓRIO DE FINAL: IDENTIFICADOR DE SINAIS TRIFÁSICOS

Rio Claro, São Paulo
2018

Leonardo Mendonça de Araujo,
Lucas Bagatini do Nascimento,
Mário Muramatsu Júnior

RELATÓRIO DE FINAL: IDENTIFICADOR DE SINAIS TRIFÁSICOS

Monografia apresentada ao curso de Ciências da Computação, como requisito parcial para a obtenção do Título de Bacharel em Ciências da Computação, Instituto de Geociências e Ciências Exatas da Universidade Estadual Paulista

Orientador: Mario Roberto da Silva

Rio Claro, São Paulo
2018

Sumário

	Sumário	3
	Lista de ilustrações	5
1	INTRODUÇÃO	7
2	MATERIAIS E MÉTODOS	8
3	DISCUSSÃO E ATIVIDADES	9
3.1	Emulador RST	9
3.2	Medidor de Período	11
3.3	Correção de Erro aleatório	12
3.4	Gerador	14
3.5	Comparador de Fase	16
3.5.1	Máquina sequencial CompFase	16
3.5.2	Circuito CompFase	17
3.6	Timeout	18
3.6.1	Circuito contador	19
3.6.2	Circuito apresentador do Timeout	20
3.7	Anti bouncing	21
4	RESULTADOS E DISCUSSÃO	24
4.0.1	Teste - Circuito RST	24
4.0.2	Teste - Gerador e Anti bouncing	24
4.1	Teste - Demais componentes	26
5	CONCLUSÃO	27

Lista de ilustrações

Figura 1 – Sinais RST	9
Figura 2 – Diagrama de Estado Emulador RST	9
Figura 3 – Tabela de estados e excitação dos Flip-flops Emulador RST	10
Figura 4 – Circuito do Emulador RST	10
Figura 5 – Circuito do Emulador RST com saídas a 60Hz	11
Figura 6 – Diagrama esquemático do medidor de período da fase de referência	11
Figura 7 – Unidade X2	12
Figura 8 – Entrada exatamente antes da borda de subida	13
Figura 9 – Entrada exatamente depois da borda de subida	13
Figura 10 – Distribuição de probabilidade do erro	14
Figura 11 – Distribuição de probabilidade do erro com centro em 0	14
Figura 12 – Diagrama de estados da nova UC	15
Figura 13 – Diagrama de estados da nova UC	16
Figura 14 – Diagrama de estados da comparadora de fases	17
Figura 15 – Circuito CompFase	17
Figura 16 – Circuito de contagem do timeout	19
Figura 17 – Circuito do display do Timeout	20
Figura 18 – Diagrama de estado do diplay do Timeout	21
Figura 19 – Fenômeno Anti bouncing	21
Figura 20 – Diagrama de estado da Unidade Anti bouncing	22
Figura 21 – Circuito da Unidade de controle com Anti bouncing	23
Figura 22 – Simulação funcional com o emulador de sinais RST	24
Figura 23 – Circuito de testes do Gerador e Anti bouncing	25
Figura 24 – Simulação funcional do Gerador e Anti bouncing - Parte 1	25
Figura 25 – Simulação funcional do Gerador e Anti bouncing - Parte 2	26
Figura 26 – Simulação funcional do Gerador e Anti bouncing - Parte 3	26

1 Introdução

2 Materiais e Métodos

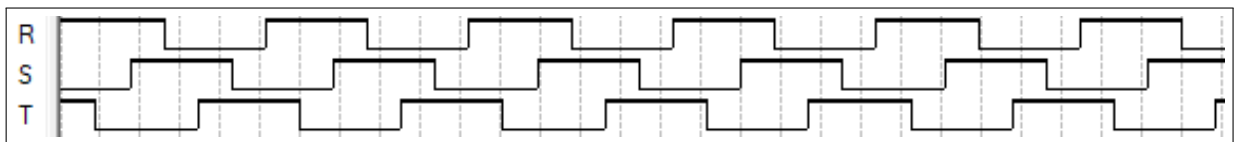
3 Discussão e Atividades

3.1 Emulador RST

O primeiro passo para começarmos a desenvolver o projeto era criar uma representação do sinal trifásico, para que este sirva de referência para os testes futuros.

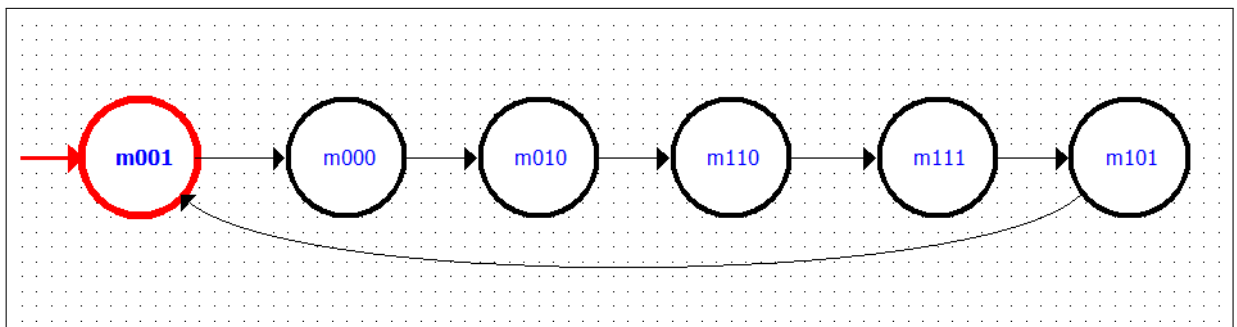
A princípio, sabemos que os sinais trifásicos possuem defasagem de 120° entre cada onda e oscilam a 60Hz, sendo assim, é necessário criar uma máquina sequencial que seja capaz de gerar tal sinal.

Figura 1 – Sinais RST



O primeiro passo foi elaborar um diagrama de sequência que satisfaça as condições para a geração dos sinais.

Figura 2 – Diagrama de Estado Emulador RST



Para a construção deste diagrama foi utilizado três bits para referenciar os estados, note que de um estado para outro apenas um bit é alterado, foi feita esta escolha pelo fato de que com isso a implementação do circuito iria ser simplificada. Montando a tabela de próximos estados e de excitação dos flip-flops.

Utilizando Karnaugh para fazer as simplificações e encontrar as expressões de cada saída do circuito, encontramos que:

$$DR = Q1$$

$$DS = \overline{Q0}$$

$$DT = Q2$$

Figura 3 – Tabela de estados e excitação dos Flip-flops Emulador RST

Estado Atual			Próximos Estados			Saídas			Flip-flops		
Q2	Q1	Q0	Q'2	Q'1	Q'0	R	S	T	DR	DS	DT
0	0	1	0	0	0	1	0	1	0	0	0
0	0	0	0	1	0	1	0	0	0	1	0
0	1	0	1	1	0	1	1	0	1	1	0
1	1	0	1	1	1	0	1	0	1	1	1
1	1	1	1	0	1	0	1	1	1	0	1
1	0	1	0	0	1	0	0	1	0	0	1
0	1	1	X	X	X	X	X	X	X	X	X
1	0	0	X	X	X	X	X	X	X	X	X

As saídas serão:

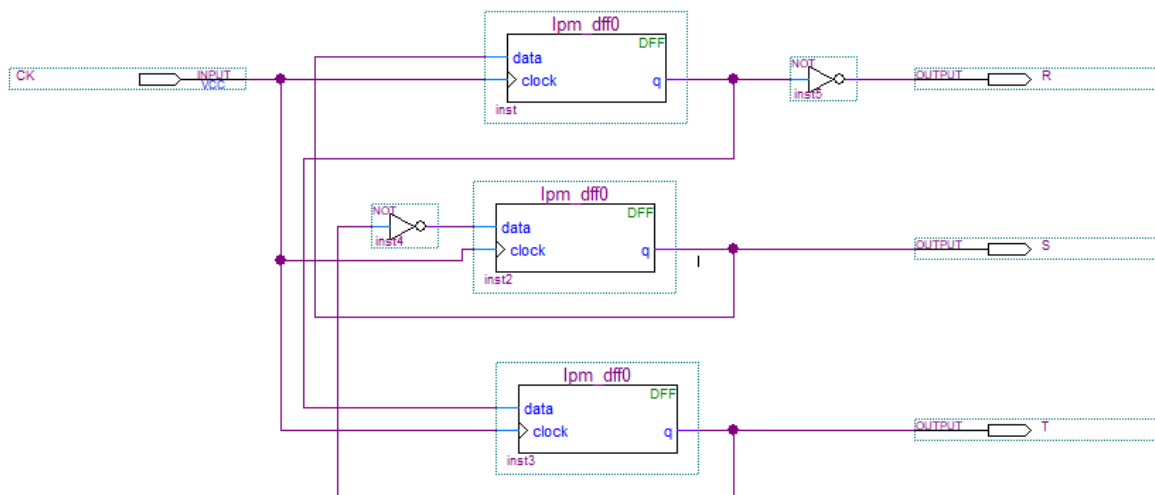
$$R = \overline{Q_2}$$

$$S = Q_1$$

$$T = Q_0$$

A partir disto podemos construir o circuito do emulador RST, representado pela Figura 4.

Figura 4 – Circuito do Emulador RST



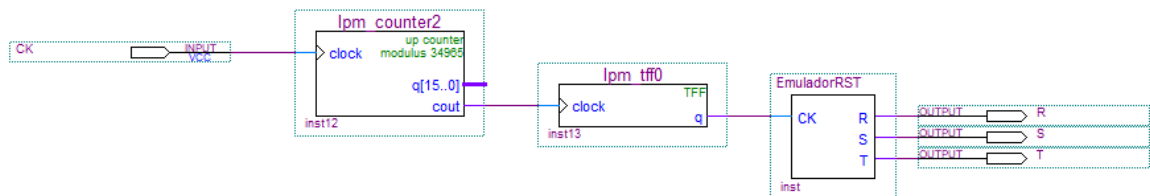
Tal circuito está gerando sinais periódicos defasados em 120° , porém apenas com isto não é possível que elas oscilem a 60Hz, pois a placa em que foram feitos os teste possui clock de 25.175MHz, portanto, é necessário reduzir o clock de entrada de tal circuito.

Para fazer tal redução foi utilizado a saída Carry-out de um contador de módulo 34965 como o clock do nosso circuito RST, com isso o clock será dividido pelo módulo do contador. Mas isso ainda não é suficiente para alcançar os 60Hz, para atingir o valor

correto foi inserido um flip-flop do tipo T para dividir o clock por dois, fazendo com que entre no circuito um sinal de 360Hz que será dividido por seis ao passar pelo emulador RST, devido ao fato desta ter 6 estados. Assim teremos os sinais RST a 60Hz.

A Figura 5 representa o circuito que gerará os sinais corretamente.

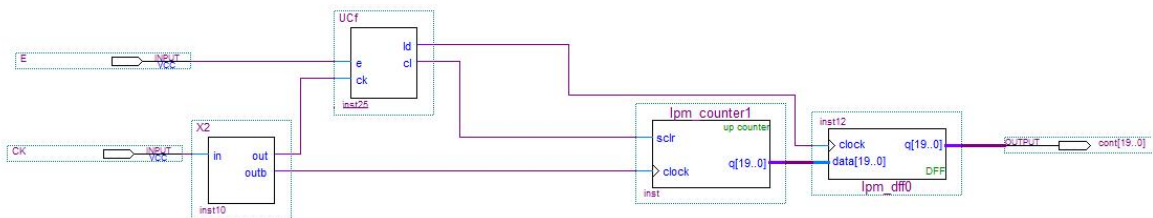
Figura 5 – Circuito do Emulador RST com saídas a 60Hz



3.2 Medidor de Período

Primeiramente precisamos ser capazes de receber um sinal de entrada e interpreta-lo para que seja possível dizer qual das três fases foi inserida, então é necessário criar um circuito capaz de identificar qual é o período da fase inserida, representado pela Figura 6.

Figura 6 – Diagrama esquemático do medidor de período da fase de referência



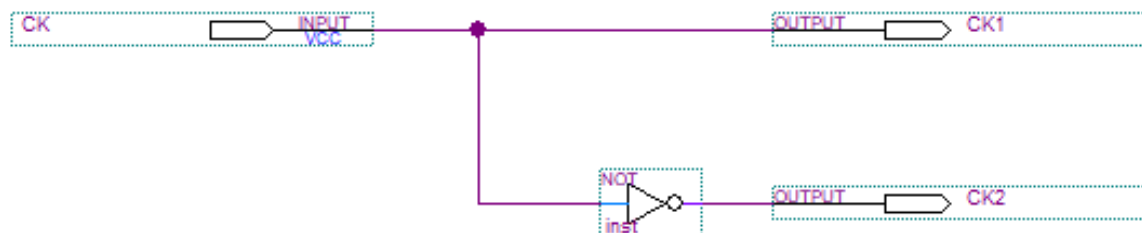
Sabendo que o período é toda a extensão desde sua primeira borda de subida até a sua próxima borda de subida, então, para conseguir identifica-lo, foi utilizado um contador que irá contar o número de clocks dentro de um período e então armazena-lo em um registrador.

Ao inicio de cada período o contador é zerado através do sinal Cl da unidade de controle UCf, tal circuito será explicada mais a frente. no mesmo clock, através do sinal Ld o ultimo valor contado é transferido para o registrador, para que ambos comandos sejam executados em ordem correta foi criado a unidade X2.

A única função da unidade da Figura 7 é, a partir de uma entrada, no caso o clock, gerar duas saídas, uma idêntica ao sinal de entrada e a outra com o sinal negado.

Note que, o clock do contador provem do sinal CK2 de X2, ou seja, tal contador trabalha na borda de descida do clock original, devido a isto os comandos Cl e Ld podem

Figura 7 – Unidade X2



ser feitos no mesmo clock, uma vez que, assim que forem gerados sinais altos de Cl e Ld, na borda de subida do clock, o valor contado pelo contador é salvo no registrador e no borda de descida deste mesmo clock, é feito o clear do contador.

3.3 Correção de Erro aleatório

Utilizando a unidade X2 para sincronizar os comandos faz com que exista a geração de um erro aleatório, possuindo dois casos extremos.

- Exite o caso em que o sinal de entrada é inserido exatamente antes da borda de subida do clock, então a zeragem do contador possuirá um atraso de 0,5 clock, ou seja, um erro de -0,5 clock, porém note que no valor registrado é desconsiderado 0,5 clock, com isto é gerado mais um erro de -0,5 clock, mostrado na Figura 8;
- No caso em que o sinal de entrada é inserido exatamente depois da borda de subida do clock, o contador será zerado com um atraso de 1,5 clocks, ou seja, com um erro de -1,5 clocks. Além disto o valor é registrado com 1 clock a mais, gerando um erro de +0,5 clock na leitura, representado pela Figura 9.

Analisando todas as combinações de erros possíveis nos dois casos extremos e utilizando a operação de convolução para estimar a probabilidade conjunta dos dois eventos (carga e zeragem), então, pode-se concluir que há a geração de um erro aleatório do qual varia de acordo com uma distribuição de probabilidade contínua com centro em -1 e bordas em -2 e 0. A Figura 10 representa tal distribuição.

Note que, pelo fato da distribuição ter centro em -1, então o erro mais provável é o de -1 clock, ou seja, a cada leitura será perdido um clock da fase de entrada, fazendo com que as medições utilizem dados incorretos. O ideal nesta situação é que o centro da curva de distribuição de probabilidade seja exatamente em 0, assim o erro mais provável será de 0 clocks, tendo assim uma leitura mais correta da fase de entrada.

Figura 8 – Entrada exatamente antes da borda de subida

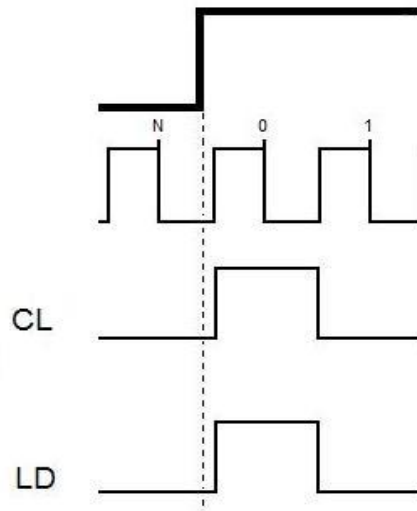
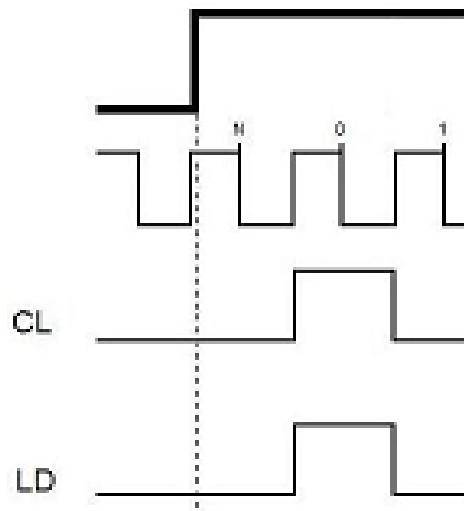


Figura 9 – Entrada exatamente depois da borda de subida



No primeiro circuito, o contador possuía uma entrada sclr (Synchronous Clear), este comando permitia zerar todos os bits do contador, porém, para corrigir tal problema alterou-se esta entrada para um sset (Synchronous set), este comando irá inserir um valor preestabelecido, no caso, será inserido o valor 1.

Desta forma iremos inserir um erro de +1 clock na leitura, uma vez que o primeiro clock não será contado, e sim inserido, porém, isso faz com que a distribuição de probabilidade atinja o comportamento ideal, ou seja, somando 1 ao erro aleatório o centro da curva ficará em 0, representado pela Figura 11.

Note que esta solução não corrige todo o problema apenas diminui a probabilidade de erros, o ideal seria que não houvessem erros na leitura. Para que seja possível garantir que o sinal lido é confiável foi criado um outro circuito que visa promover isto, tal circuito será apresentado mas a frente.

Figura 10 – Distribuição de probabilidade do erro

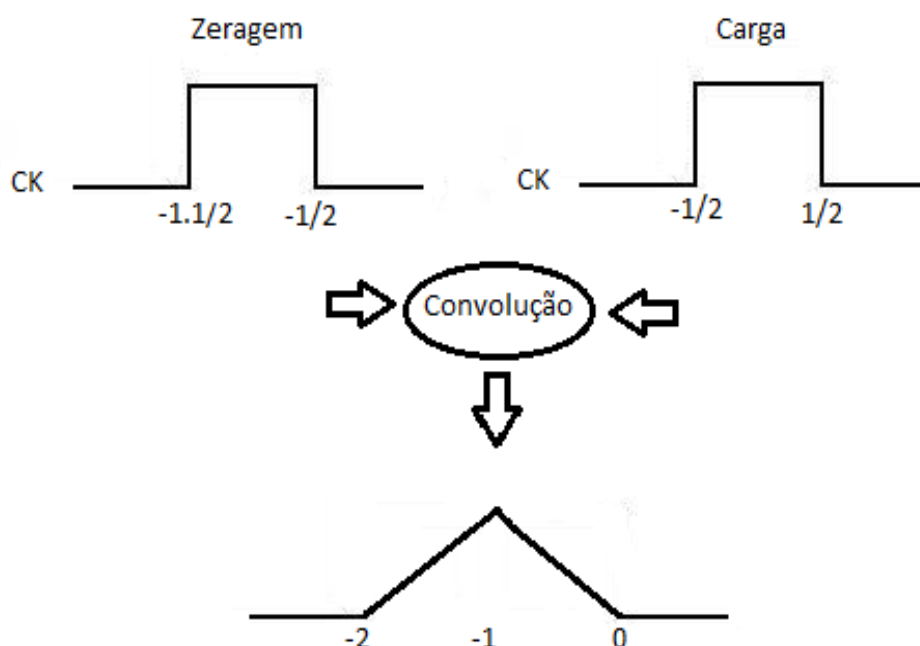
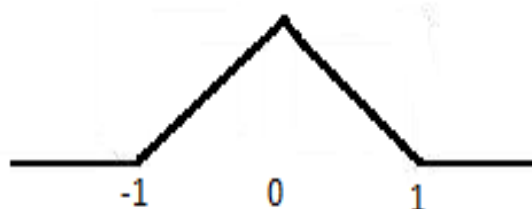


Figura 11 – Distribuição de probabilidade do erro com centro em 0



3.4 Gerador

O circuito atual é capaz de medir o período de um sinal de entrada, porém, tudo será perdido se o sinal for retirado, o próximo passo então é criar um circuito capaz de gerar um sinal em sincronismo com o sinal de referência recebido.

Primeiramente foi construído um diagrama de estado para uma UC (Unidade de Controle) capaz de gerenciar os componentes do circuito com o objetivo de fazer a cópia do sinal recebido, representada pela Figura 12.

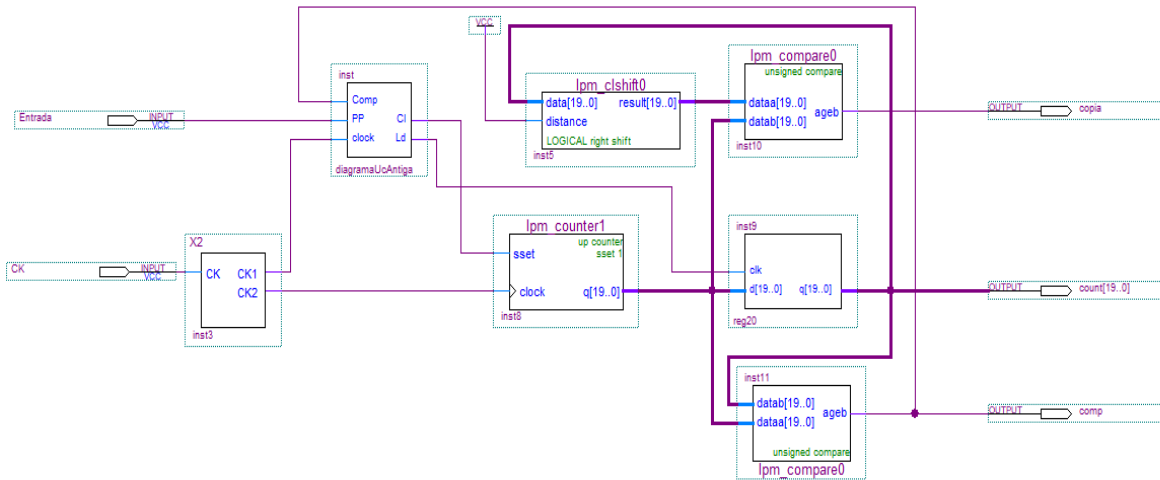
Ao conectar um sinal de entrada ao circuito, este começará no estado 1 e então irá para o estado 2 quando PP (sinal de entrada) estiver em 1. Para passar para o estado 3 é necessário que PP esteja em 0. Note que este processo é feito para ignorar o primeiro

Chegando ao estado 4, o sinal Cl ficará em nível alto, limpado assim o contador, em seguida passa-se para o estado 5, em que, junto com o estado 6 irão esperar passar outro período, chegando finalmente no estado 7, em que são gerados sinais Cl e Ld. Garantida esta primeira medição correta do período, a máquina evolui sucessiva e repetidamente pelos estados 8, 9 e de volta para o 7, onde são providos os sinais Cl e Ld, permitindo a realização de novas e corretas medições, até que o sinal de entrada for retirado e o sinal Comp esteja em 1.

Indo para o estado 10, onde é gerado um sinal Cl para manter o sincronismo com a fase lida e então passa para o estado 1. A partir deste ponto o a maquina fica repetidamente indo do estado 1 para o 10, esta transição irá ocorrer sempre que o sinal de entrada estiver em 0 e o Comp estiver em alto, ou seja, sempre ao fim de um período.

Outra mudança necessário foi a troca do registrador (inst9) para um que começasse com o valor máximo possível, ou seja, todos os bits em 1. Isto se deve pois, como o lpm_counter1 (inst 8) começa sempre em 1, e se o comparador começar em 0, o sinal Comp não será ativado corretamente.

Figura 13 – Diagrama de estados da nova UC



3.5 Comparador de Fase

Após inserir um sinal de entrada, o circuito é capaz de gerar uma cópia desta fase, porém, é impossível descobrir qual fase foi inserida (R, S ou T), para isso, foi implementado três botões (Rin, Sin e Tin) que devem ser pressionados para indicar qual fase foi inserida inicialmente.

Ao pressionar um dos botões de entrada, a saída LedOutput deve ser equivalente ao botão pressionado, isso acontece até que se tire o dedo do botão. Assim quando chegar um novo sinal de Clear, não existirá mais sinal desses botão, contudo a saída deve continuar a mesma, isto porque a lógica do componente saídaLed é realimentada com os valores de Rout, Sout e Tout. Apenas terá alteração na saída ao colocar um novo sinal na ponta de prova.

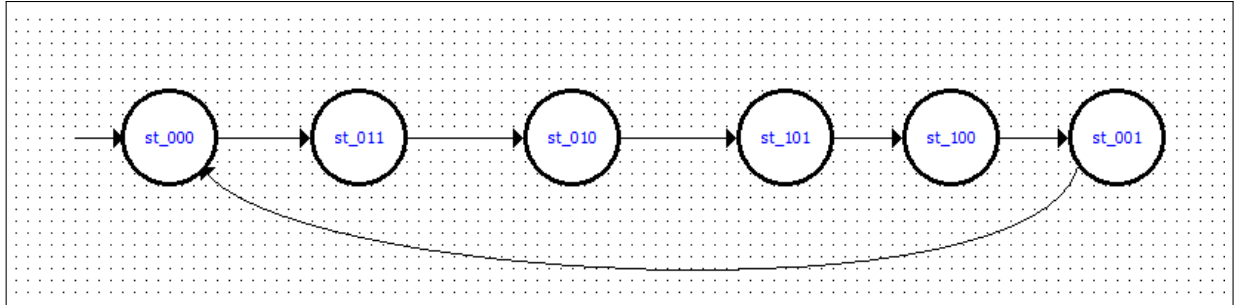
3.5.1 Máquina sequencial CompFase

Neste ponto, a máquina está pronta para executar a leitura de uma nova fase para fazer a comparação. A máquina sequencial da Figura 14 sinaliza se o novo sinal de entrada está igual a fase de referencia, se é a fase anterior (na sequencia RST) ou a fase posterior, para isto utiliza a seguinte lógica:

- estados começando em 00, indicam que a fase inserida é igual a fase de referencia;
- estados começando em 01, indicam que a fase inserida é a posterior à fase de referencia;
- estados começando em 10, indicam que a fase inserida é a anterior à fase de referencia.

Perceba que o ultimo bit serve para diferenciar os estados cujos valores de saídas são iguais.

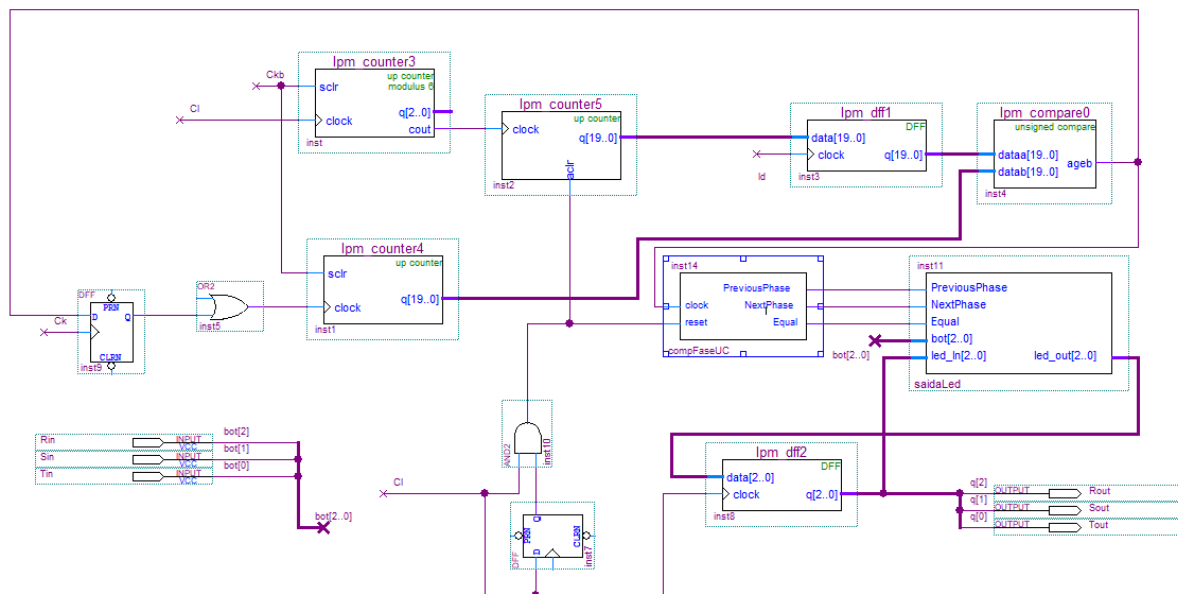
Figura 14 – Diagrama de estados da comparadora de fases



3.5.2 Circuito CompFase

A partir da máquina sequencial apresentada, pode-se construir o circuito da Figura 15.

Figura 15 – Circuito CompFase



Para utilizar o método de comparação de fases requer uma frequência de seis vezes a frequência da fase de referência. Assim, o contador `Ipm_counter3` (inst1) gera esta frequência dividindo o clock por seis, sendo então utilizado para realizar uma contagem entre as bordas de subida da fase de referência pelo contador `Ipm_counter5` (inst2), resultando em uma contagem truncada em $\frac{1}{6}$ do período da fase de referência, sendo armazenada em um registrador `Ipm_dff1` (inst3) pelo pulso `Ld`.

Como o contador `Ipm_counter5` (inst2) recebe um clock com $\frac{1}{6}$ da frequência original do circuito, é utilizado um Clear assíncrono que ocorre na borda de descida anterior ao

pulso Cl, com metade de sua duração, proveniente da porta "E"AND2 (inst10) entre o pulso Cl e o flip-flop tipo D DFF (inst9), que também realiza o Clear na máquina comparadora de fases.

O comparador Ipm_compare (inst4) provém então o clock necessário para a máquina comparadora de fases, e realiza também o Clear do contador Ipm_counter4 (inst1), através da comparação entre a contagem truncada em $\frac{1}{6}$ do período da fase de referência e a do próprio contador, após o atraso de meio período de clock, obtido com o flip-flop tipo D DFF (inst9), pois sua saída também ocorre em uma borda de descida. Como uma divisão por seis pode envolver resto, o contador Ipm_counter4 (inst1) também recebe outro sinal de Clear através da porta "OU"OR2 (inst5) produzido a cada período da fase de referência, evitando-se o acúmulo de erros.

3.6 Timeout

Na seção 3.3, em que corrigimos o problema de erro aleatório na leitura da fase de referência, foi dito que aquilo era o suficiente para acabar com os erros de leitura, uma vez que, os erros variam de acordo com uma distribuição de probabilidade continua com centro em 0, isto é, pode variar de -1 a +1 clock de erro.

Este valor pode parecer pequeno, mas este erro se repete a cada novo período enquanto se espera a conexão da nova fase de referência à ponta de prova. O sinal gerado internamente se defasa em relação a fase de referência para mais ou para menos.

Se o acumulo deste erro se torna muito grande então a comparação pode não ser tão precisa, pois, no caso em que a fase R é inserida como fase de referencia, após a cópia se defasar em $\frac{1}{6}$ para menos em relação a fase de referencia e inserindo novamente uma fase R para fazer a comparação, tal situação se confunde com o caso em que é inserido a fase S para fazer a comparação e a cópia se defasou em $\frac{1}{6}$ para mais.

Para corrigir isto criou-se o circuito Timeout, que nada mais é do que um "temporizador". Este circuito ira executar uma contagem decrescente, até que a fase de referencia esteja defasada (situação limite é quando esta atinge $\frac{1}{6}$ de erro). Ao atingir o valor 0, a máquina será travada, indicando que o sinal copiado não é mais confiável, necessitando de uma nova leitura para a fase de referencia.

O tempo deste Timeout pode ser estimado, sendo sua fórmula do tipo:

$$Timeout = \frac{Clock}{(6 \cdot (frequencia)^2)}$$

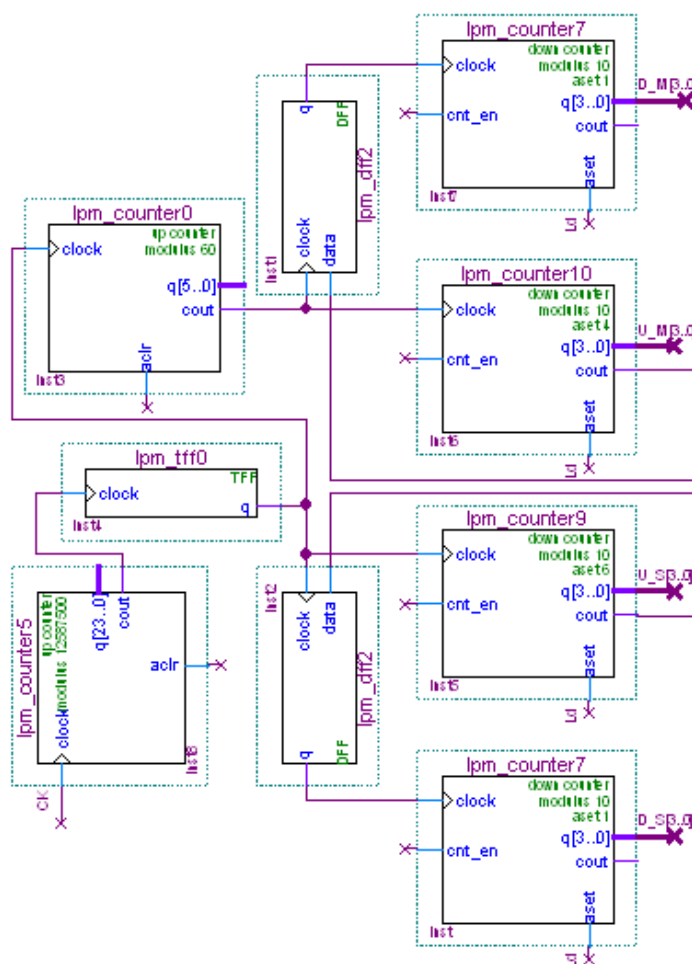
Sendo o clock da placa igual a 25175MHz e a frequência da rede entre 60Hz a 70Hz, temos o tempo aproximadamente de 856s, equivalente a 14 minutos e 16 segundos.

3.6.1 Circuito contador

Para fazer a contagem, construiu o circuito da Figura 16. O contador `lpm_counter5` (inst8) irá dividir o clock de 25175MHz por dois, gerando um sinal de 2Hz que será dividido novamente por 2 pelo flip-flop tipo T gerando um sinal de 1Hz. Tal sinal é então dividido pelo contador de módulo 60 (inst3) produzindo um pulso a cada minuto. Os contadores à direita irão receber estes sinal, sendo que os dois superiores executam a contagem em segundos e os inferiores a contagem em minutos, o cout dos contadores de unidades servem para acionar os contadores de dezenas.

O valor inicial calculado acima é inserido através dos pinos aset (Asynchronous Set) dos contadores.

Figura 16 – Circuito de contagem do timeout

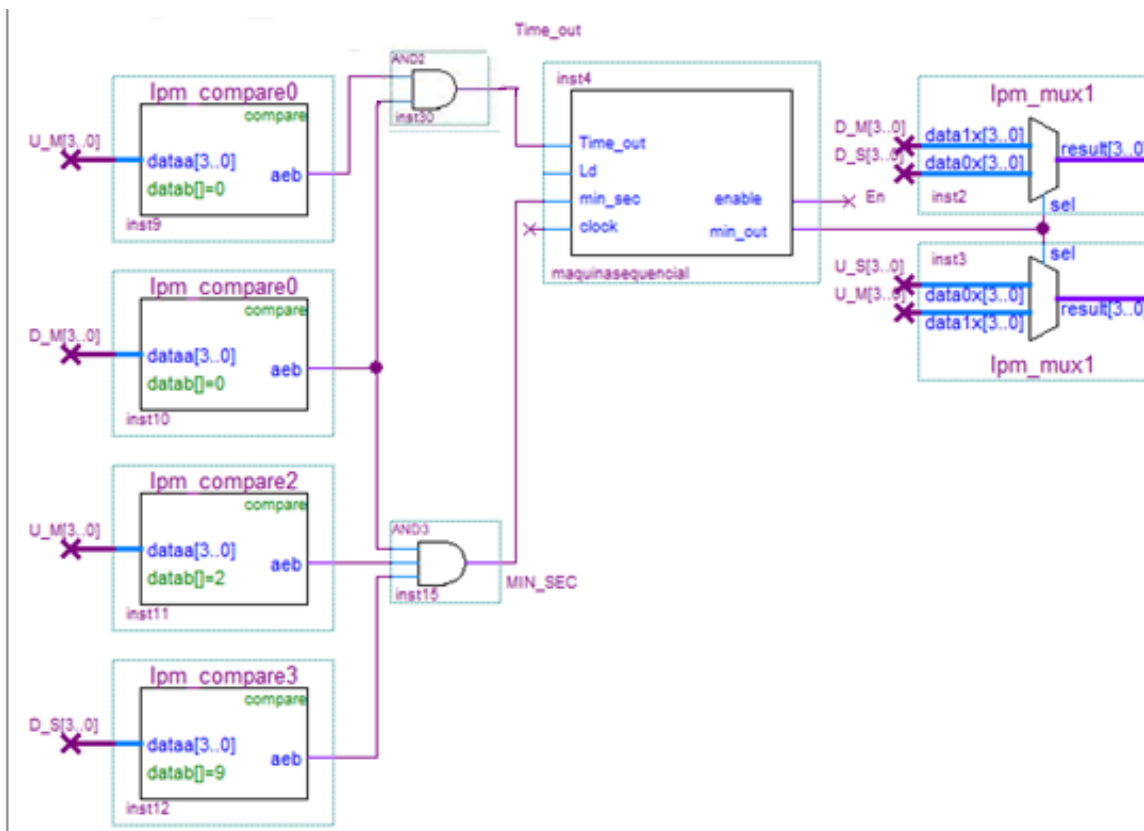


Quando tais contadores de tempo em minutos alcança 00 é produzido o sinal que deverá travar a máquina toda até que nova fase seja colocada na ponta de prova e esta seja identificada pelo pressionar de um dos botões.

3.6.2 Circuito apresentador do Timeout

Ao atingir o valor de 2 minutos, o contador de segundo deverá estar com tempo restante de 20 segundos, após passar este tempo, a contagem passará a mostrar apenas o valor em segundos, sendo o que este começará em 99 segundos. A combinação entre o valor 2 dos minutos com o 9 dos segundos gera o sinal MIN_SEC que irá alterar a contagem apenas para segundos. O circuito que faz tão função é representado pela Figura 17

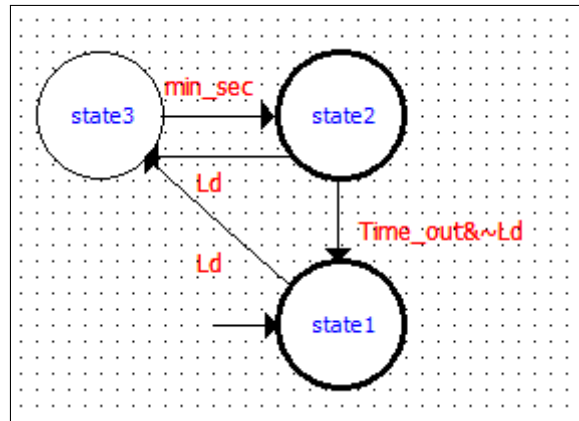
Figura 17 – Circuito do display do Timeout



O bloco lógico Time_out é representado pelo seguinte diagrama de estado, Figura 18. O circuito irá para o estado 3, onde a contagem de minutos esta ativa, nos estados 1 e 2, a contagem de minutos é desativada. Quando a máquina está nos estados 2 e 3, os pinos enable dos contadores estarão ativos, liberando os contadores para continuarem a contar, porém, no estado 1 estes enables são desativados, travando a contagem.

A transição do estado 3 para o 2 ocorre quando o sinal MIN_SEC estiver em 1, ou seja, quando a contagem atinge 99 segundos. No estado 2, caso a ponta de prova seja conectada antes do termino do tempo de timeout, a máquina voltará para o estado 3, do contrario ele irá para o estado 1, onde irá ficar travada até que aja uma nova leitura na ponta de prova.

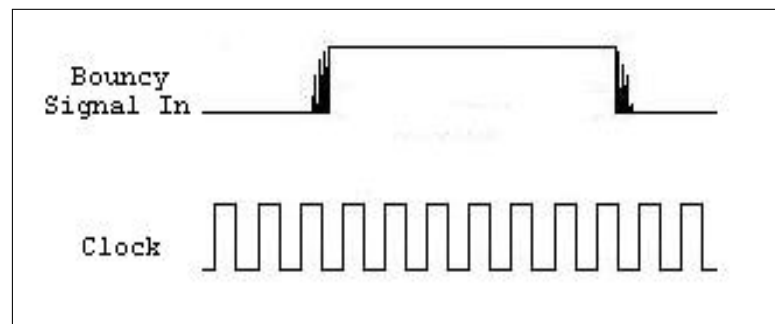
Figura 18 – Diagrama de estado do diplay do Timeout



3.7 Anti bouncing

Ao conectar a ponta de prova ao sinal de referencia ocorre um fenômeno chamado de bouncing, que consistem em oscilações aleatórias geradas pelo contato mecânico entre o fio e a ponta de prova. Isto ocorre devido à condição física dos conectores, uma vez que estes possuem uma superfície rugosa causando uma instabilidade inicial, que provocam pequenas oscilações. Este fenômeno é representado pela Figura 19

Figura 19 – Fenômeno Anti bouncing

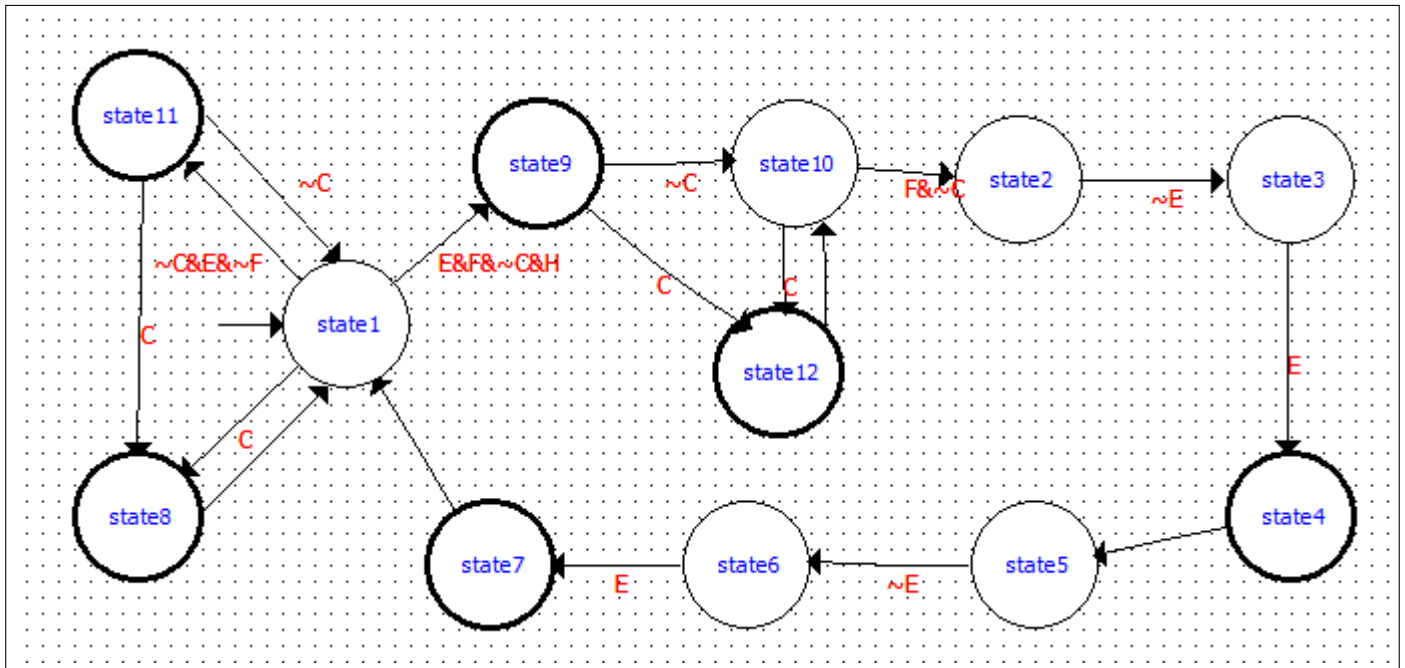


Tais oscilações podem influenciar nos resultados, fazendo com que a máquina sequencial transite pelos estados sem ter um sinal confiável.

Para corrigir este problema, foi proposto a seguinte máquina sequencial (Figura 20) que servirá como Unidade de controle do gerador da cópia. Basicamente ele irá travar a leitura até que se passe um tempo necessário para as entradas se estabilizarem.

Ao iniciar a máquina, ela permanecerá no estado 1 até que uma ponta de prova seja conectada ao dispositivo e o tempo característico de bouncing já tenha passado e o sinal H, que é recebido através do pressionar de um dos botões que indicam qual é a fase inserida, esteja em 1. Se estas condições forem cumpridas então a máquina evoluirá para o estado 9, onde produzirá um sinal Rf (Reset Flag), por um clock, e irá para o estado 12 caso o sinal Comp esteja em 1, do contrário irá para 10. O caso em que Comp estiver em 1 significa que houve um transcurso de mais um período registrado, para isto, é necessário

Figura 20 – Diagrama de estado da Unidade Anti bouncing



então que o sinal Cl esteja em 1, Cl é colocado em nível alto, por um clock, no estado 12 e em seguida ele irá para o estado 10.

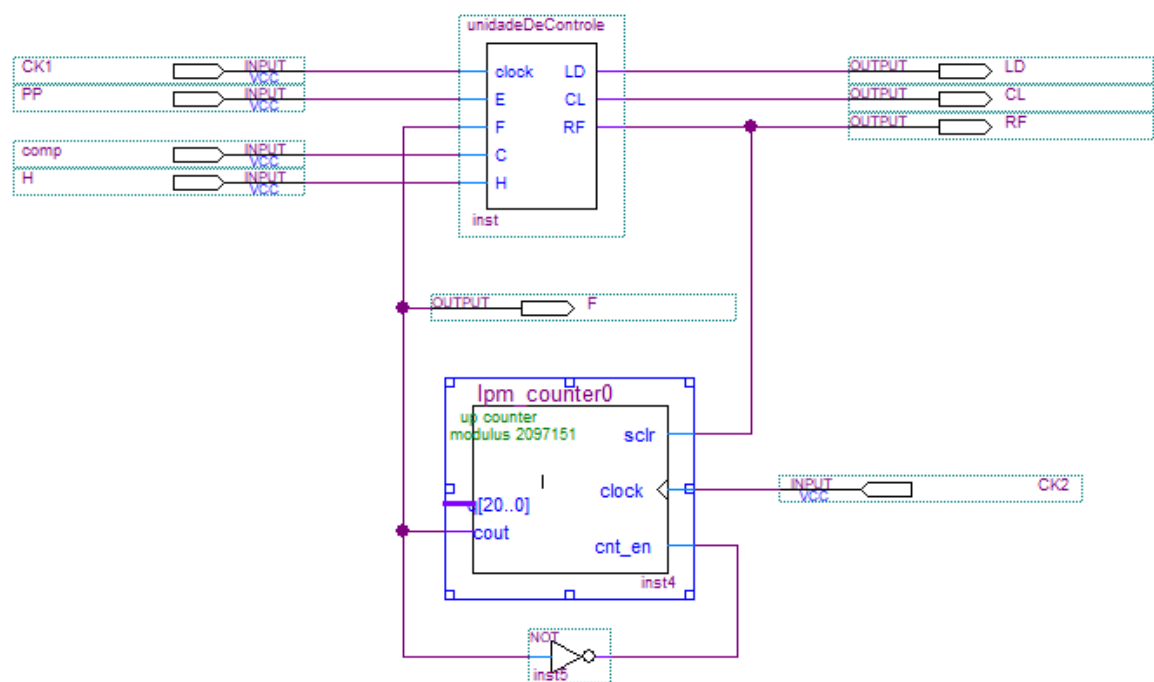
A máquina ficará no ciclo entre o estado 10 e 12 até que o sinal F esteja em 1, ou seja, até passar o tempo característico de bouncing. Os estados 2 e 3 irão permitir a detecção da próxima borda de subida da fase. No estado 4, é gerado um sinal Cl, que irá zerar o contador de período, no estado 5 e 6 seja possível fazer a contagem do período corretamente e então salvando-o no estado 7 com um sinal Ld e outro Cl, para finalmente retornar ao estado 1.

Neste ponto, sempre que o sinal Comp estiver em 1 a máquina irá para o estado 8, em que será gerado um sinal Cl para manter o sincronismos com a fase de referencia. Caso a ponta de prova ainda esteja conectada e em nível alto, a máquina irá para o estado 11, para gerar um sinal Rf para impedir que F seja ativada, e logo em seguida irá voltar para o estado 1 ou primeiro irá para o estado 8, pela necessidade de um clear e em seguida retornará ao 1.

A partir deste diagrama foi construído o circuito da Figura 21.

O sinal Rf que é gerado para não permitir que a entrada F esteja em 1 nada mais é do que um sclr (Synchronous clear) do lpm_counter (inst14). Este contador irá marcar o tempo característico de bouncing, ao final desta contagem, o sinal cout (Carry out) será expelido, indicando que o contador atingiu o seu limite, ou seja, passou-se o tempo necessário. Note que esta saída influencia o próprio contador, uma vez que o sinal cout é inserido na entrada cnt_en (Counter Enable). Ao inseri-la, o contador será desativado, até que seja dado um clear.

Figura 21 – Circuito da Unidade de controle com Anti bouncing

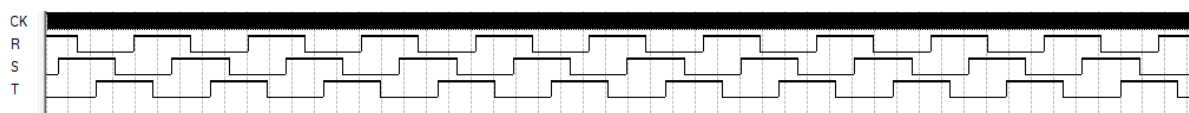


4 Resultados e Discussão

4.0.1 Teste - Circuito RST

O primeiro circuito implementado foi o gerador de sinais RST, pois este serviria para fazer todos os testes seguintes. Para que fosse possível visualizar a variação das ondas geradas pelo circuito, foi necessário diminuir o valor do contador de `lpm_counter2` (inst12) da Figura 5, pois este estaria variando a 60Hz, algo que dificultaria a visualização. Alterando o valor para 10 chegamos aos resultados da Figura 22.

Figura 22 – Simulação funcional com o emulador de sinais RST



Assim como o esperado o circuito é capaz de gerar três ondas defasadas em 120° , podendo concluir que tal circuito apresenta uma representação próxima ou até mesmo fiel à situação real.

Este circuito foi testado na placa utilizando os displays de 7 seguimentos presentes nela. Cada sinal RST era representado por um seguimento, quando o sinal de uma das fases ficasse em nível alto, o seu respectivo seguimento era aceso.

4.0.2 Teste - Gerador e Anti bouncing

Para conseguir teste esta parte foi necessário assumir que o sinal H sempre estaria em 1, isto porque a parte da implementação dos botões que indicariam qual é a fase de entrada não havia sido implementada neste ponto. O sinal de entrada utilizado foi gerado pelo emulador RST e para simular o fenômeno de bouncing utilizamos um multiplexador que simularia uma instabilidade na entrada. 2777 ns O teste foi feito com o clock em 2777ns no circuito da Figura 23.

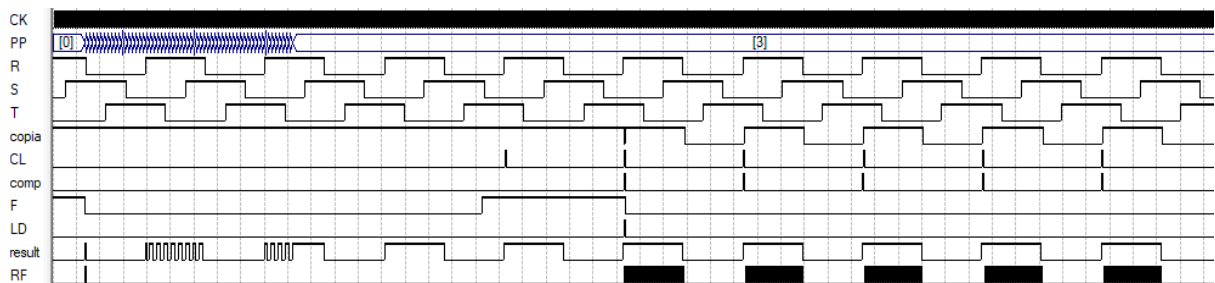
Note que a parte inferior do circuito é idêntica ao emulador RST, a única mudança é o multiplex inserido nas saídas deste. O pino de entrada PP foi utilizado para selecionar uma das fases que seriam retroalimentadas na Unidade com o anti bouncing.

Outra mudança necessário foi a alteração do módulo do contador `lpm_counter` (inst4) da Figura 21 para 400, pois do contrário o tempo característico de bouncing iria ser muito grande e então não seria possível ver o funcionamento do circuito.

Para a execução do teste foi inserido inicialmente o sinal R para que seja executado a cópia e depois que o sinal de referencia estivesse sincronizado, a ponta de prova foi colocada em nível baixo, e em seguida inserido o sinal S.

[illegible]

Figura 24 – Simulação funcional do Gerador e Anti bouncing - Parte 1



Também foi simulado a retida da ponta de prova, para isto também a simulação do fenômeno de bouncing, este teste está representado na Figura 25. Quando começamos a gerar a instabilidade na ponta de prova, vários Rf são feitos, até que PP se torna 0 e os pulsos em Rf para, porém, o pino cópia continua a gerar um sinal em sincronia com o sinal R.

Figura 25 – Simulação funcional do Gerador e Anti bouncing - Parte 2

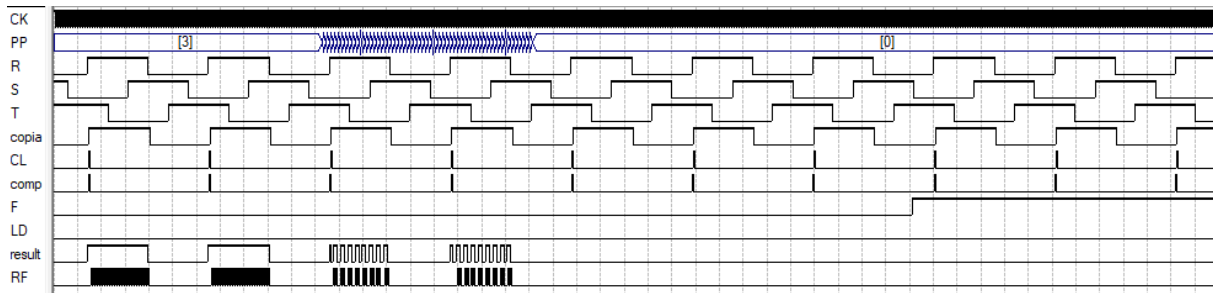
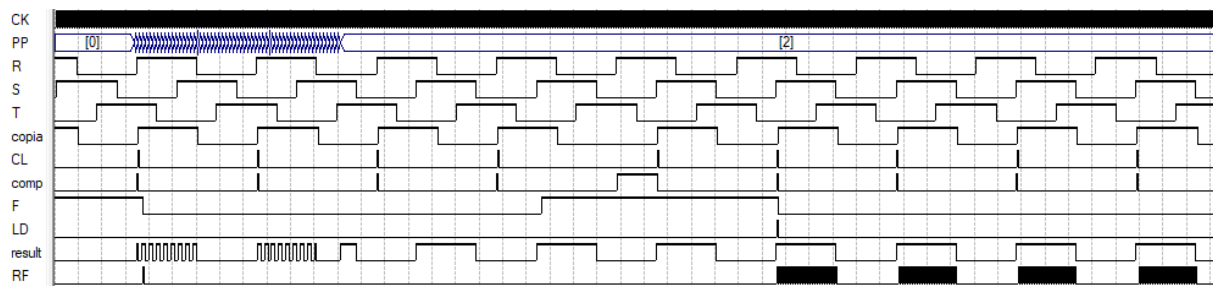


Figura 26 – Simulação funcional do Gerador e Anti bouncing - Parte 3



Ainda neste mesmo teste, simulamos a inserção de uma nova fase de referencia, representado pela Figura 26. A princípio começamos a gerar a instabilidade no sinal de PP, assim como na primeira leitura, é gerado um pulson em Rf para zerar o contador da UC e começar a contar o tempo característico de bouncing.

O fim deste tempo acontece quando F volta para o nível alto e então é gerado um sinal Cl na primeira borda de subida de S depois do tempo de bouncing. Ao fim deste período é gerado os sinais Cl e Ld, assim como na primeira leitura. Note que antes deste do primeiro Cl depois do tempo característico de bouncing o sinal de cópia ainda era igual a R, após isto, o sinal passa a ser igual ao S.

Este circuito foi testado na placa, porém não apresentou resultados positivos. Seu comportamento, quando rodando na placa, apresentava erros dos quais não fomos capazes de descobrir. Tentamos várias modificações, porém, alguns erros persistiam, outros sumiam e novos erros surgiam, impossibilitando uma conclusão precisa.

4.1 Teste - Demais componentes

Os demais componentes não foram possíveis de serem testados, devido a dúvidas de implementação, testes sem sucesso e falta de tempo.

5 Conclusão