5
Simulação e análise dos módulos desenvolvidos

Este capítulo tem como objetivo detalhar as simulações e os testes lógicos realizados através dos softwares Modelsim e ChipScope respectivamente, analisando o comportamento dos módulos desenvolvidos.

5.1.
Simulação sem a presença de retardo entre o transmissor e o receptor

Esta simulação consiste em analisar o mecanismo de alinhamento proposto no capítulo anterior verificando se ocorre ou não em algum estado o alinhamento desejado. Considerou-se primeiramente uma conexão direta entre o transmissor e o receptor, ou seja, os bits de saída do contador LFSR denominada *SERDES_TD* são também os bits de entrada do bloco ALINHAMENTO_COMMA denominado *SERDES_RD* e ambos possuem o mesmo relógio no processo de simulação realizado. O bloco abaixo contendo o desenvolvimento do módulo transmissor LFSR com o módulo receptor AVALIA_ERRO foi chamado de BER_TOP.

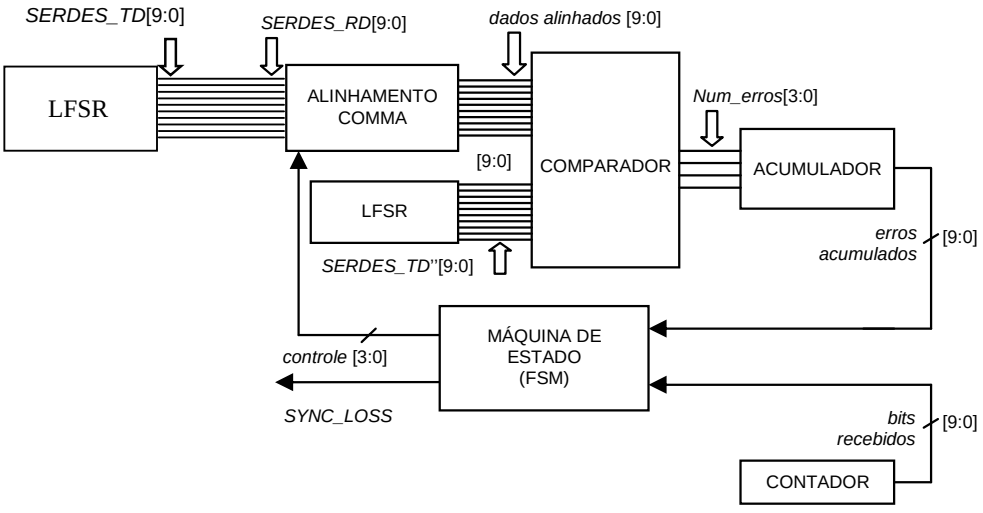

Figura 25. Simulação de uma conexão direta entre o transmissor e o receptor representando o bloco BER_TOP.

Sem a presença de retardo entre o transmissor e o receptor verificou-se que o sistema alinhou corretamente as duas seqüências no 3º estado de `alinhamento_ comma0` (`al_cm0`) como podemos observar na Figura 26 abaixo.

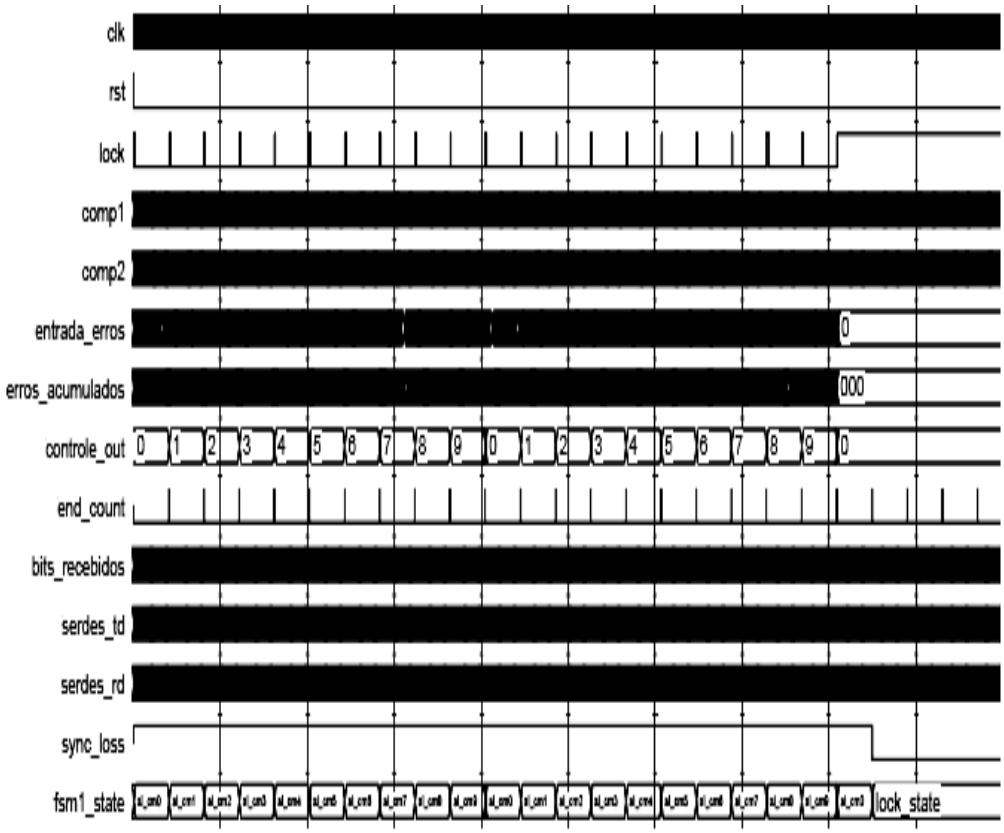

Figura 26. Simulação completa minimizada desde o início até o momento em que o sistema alinha.

A Figura 26 acima corresponde à simulação completa minimizada para podermos observar desde seu início até o momento em que o sistema alinha as seqüências *dados_alinhados* representada pelo sinal ‘comp1’ e *SERDES_TD’’* representada por ‘comp2’. Alguns sinais da Figura 26 apresentam abreviações ou equivalências em suas nomenclaturas em relação aos nomes referenciados nos blocos do capítulo anterior, mas estas serão devidamente identificadas com suas

respectivas portas de origem. Por exemplo, o sinal ‘entrada_erros’ corresponde à porta de entrada *Num_erros* do bloco ACUMULADOR ou também à porta de saída do bloco COMPARADOR; o sinal ‘controle_out’ é atribuído à porta de saída *controle* do bloco Máquina de Estados ou também à porta de entrada do bloco ALINHAMENTO_COMMA. Os sinais representados em negrito serão expandidos nas próximas figuras para podermos entender melhor o resultado das simulações.

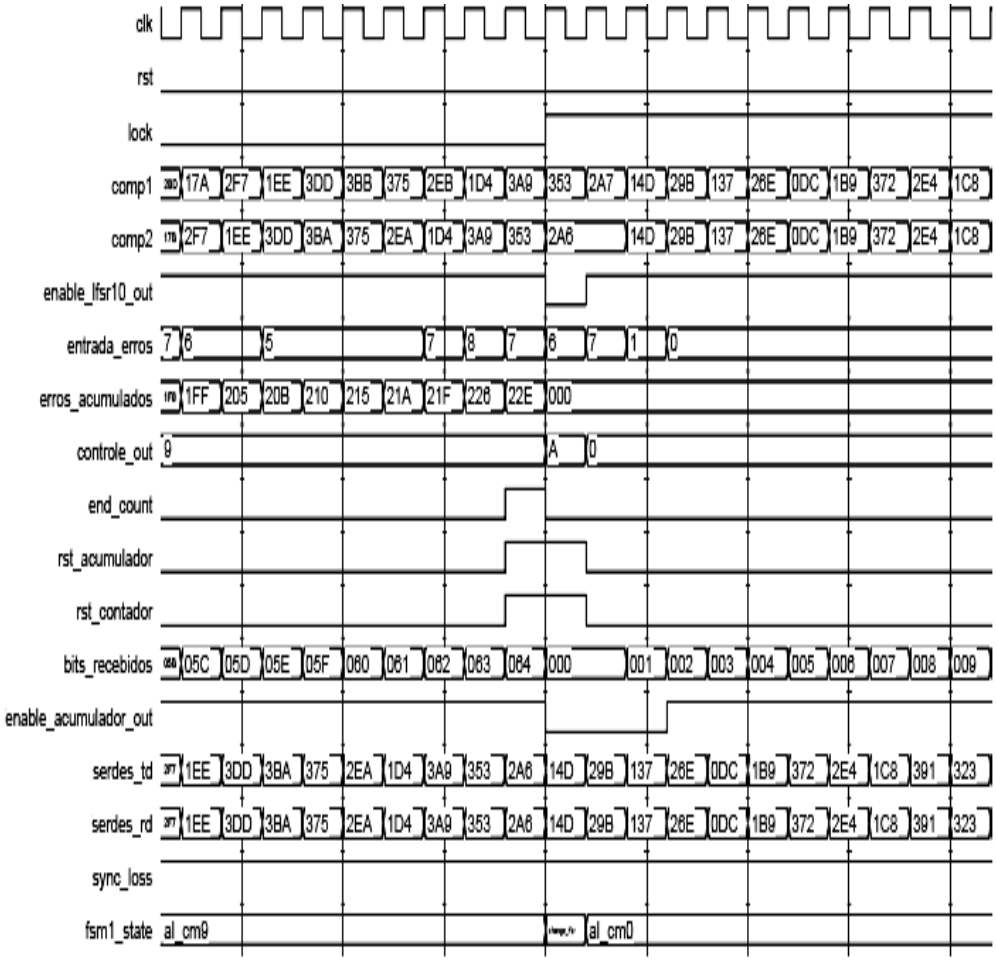

Figura 27. Simulação ampliada do momento em que os erros acumulados do sistema zeraram indicando o alinhamento correto.

A Figura 27 acima apresenta a transição entre o 2º estado `alinhamento_comma9` e o 3º estado `alinhamento_comma0`. Como podemos observar, neste estado `alinhamento_comma0` a porta de saída

erros_acumulados do bloco ACUMULADOR não apresenta nenhum erro sendo este o estado responsável pelo alinhamento do sistema. Sendo assim, as seqüências *dados_alinhados* (comp1) e *SERDES_TD"* (comp2) foram alinhadas corretamente.

O sinal 'serdes_td' representa a porta de saída *SERDES_TD* do bloco LFSR da transmissão e o sinal 'serdes_rd' representa a porta de entrada *SERDES_RD* do bloco ALINHAMENTO_COMMA da recepção. Pode-se observar na simulação que ambos os sinais são iguais, uma vez que não temos nenhum retardo entre eles e a saída do *SERDES_TD* coincide com a entrada do *SERDES_RD*.

Pode-se observar também que o sinal *lock* é habilitado no momento em que o sinal *erros_acumulados* torna-se nulo e o sinal *end_count* é habilitado quando o sinal *bits_recebidos* for igual a 100_{dec} ou 64_{hex} representando a última transição de cada estado. Nesta última transição, o contador e o acumulador são resetados.

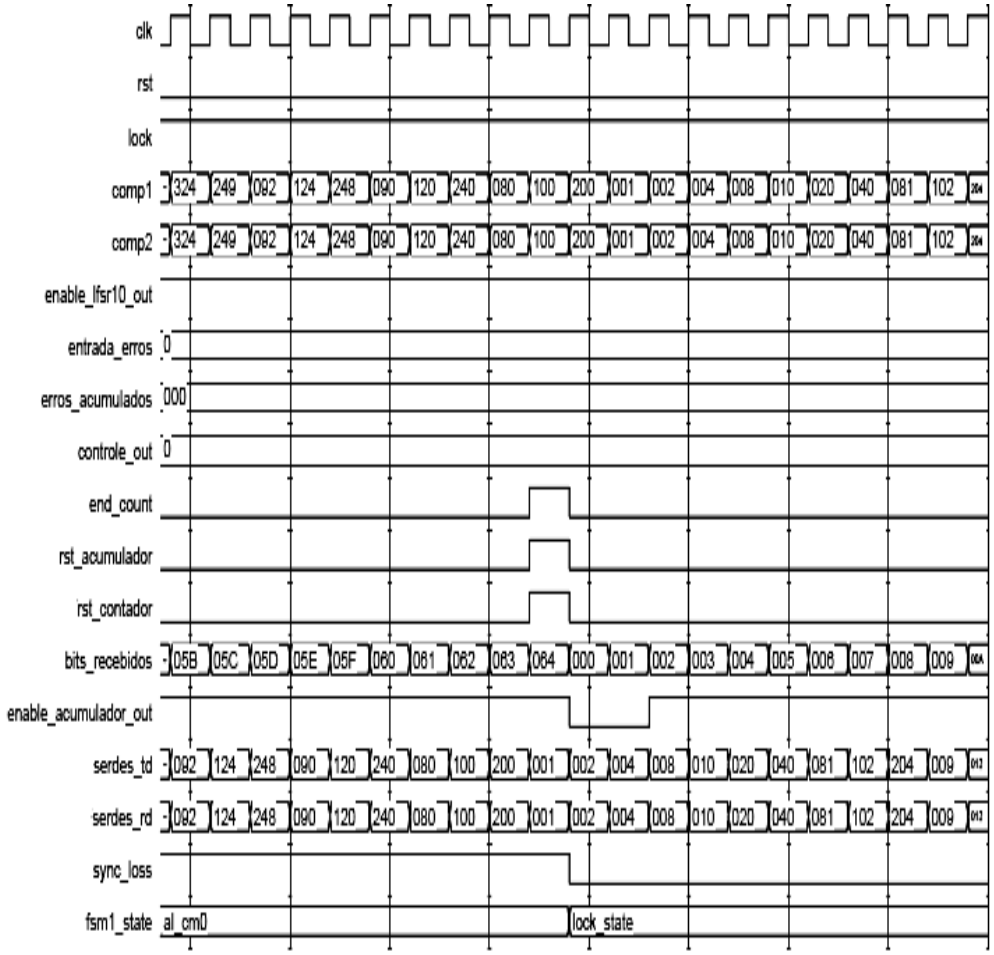

Figura 28. Simulação indicando a transição do 3º estado `alinhamento_comma0` (`al_cm0`), ao qual o sistema alinhou corretamente, para o estado `lock_state` de travamento.

Na Figura 28 acima, após as últimas 100 palavras de 10 bits do 3º estado `alinhamento_comma0`, responsável pela sincronização das seqüências, há a transição para o estado `lock_state` onde o sistema permanecerá travado com o alinhamento correto. Neste estado de travamento, podemos observar que o sinal *sync_loss* é zero.

5.2.
Simulação com a presença de retardo entre o transmissor e o receptor

Neste caso, criou-se um bloco de retardo entre o transmissor e o receptor com o intuito de simular a presença da fibra com o chip SERDES e analisar se em algum estado o sistema passaria a alinhar corretamente.

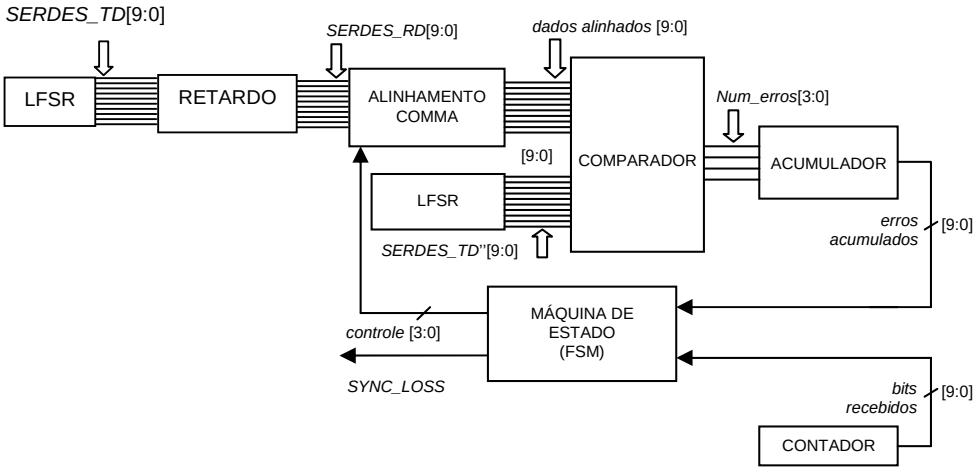

Figura 29. Simulação com a presença de retardo entre o transmissor e o receptor.

O bloco RETARDO representa o meio para a simulação feita em VHDL e apresenta três portas de entrada: *relógio*, *reset* e *SERDES_TD* (10 bits) e duas portas de saída: *relógio RBC0*[10] e *SERDES_RD* (10 bits). A presença deste bloco tem como objetivo desalinhar as seqüências adicionando um determinado atraso para verificar se posteriormente o sistema ainda assim conseguiria alinhar.

Para gerar este atraso utilizaram-se três registradores de 10 bits como apresentado na figura abaixo sendo que a cada transição positiva de relógio, o registro REG1 passa a obter os valores da seqüência *SERDES_TD* e o registro REG2 passa a obter os valores do registro REG1. Sem a dependência da transição

[10] O *relógio RBC0* (Receive Byte Clock) do chip SERDES é um recuperador de relógio usado para sincronização dos 10 bits de saída de dados do receptor deste chip (RD – Receive Data). O *relógio RBC0* é o relógio de entrada do bloco AVALIA_ERRO desenvolvido, mas na simulação considerou-se este sendo igual à porta *relógio* do bloco RETARDO.

de relógio, os valores de REG_OUT são determinados através da concatenação do bit menos significativo de REG2 com os nove bits mais significativos de REG1 e os valores da seqüência *SERDES_RD* passam a obter os valores do registro REG_OUT.

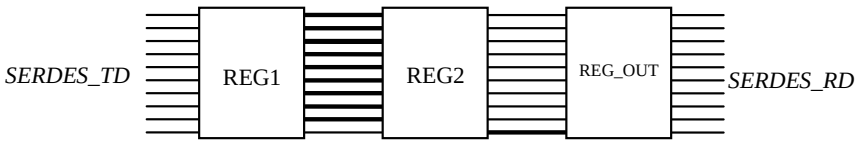

Figura 30. Representação do bloco RETARDO da Figura 29 acima.

Com a presença de retardo entre o transmissor e o receptor o sistema demorou um pouco mais para alinhar em relação ao caso do item 5.1 anterior, como era esperado e verificou-se o alinhamento correto no 5º estado de `alinhamento_comma9` (`al_cm9`) de acordo com as Figuras 31, 32 e 33 a seguir.

De forma similar ao item anterior, primeiramente apresentou-se na Figura 31 abaixo a simulação completa desde seu início até o momento em que o sistema alinha as duas seqüências. Em seguida, tem-se na Figura 32 a simulação do momento em que os erros acumulados do sistema tornaram-se nulos com a transição do estado `alinhamento_comma8` para o estado `alinhamento_comma9`, responsável pelo alinhamento correto. E por último, tem-se na Figura 33 a simulação onde ocorre a transição do estado `alinhamento_comma9` para o estado de travamento `lock_state`.

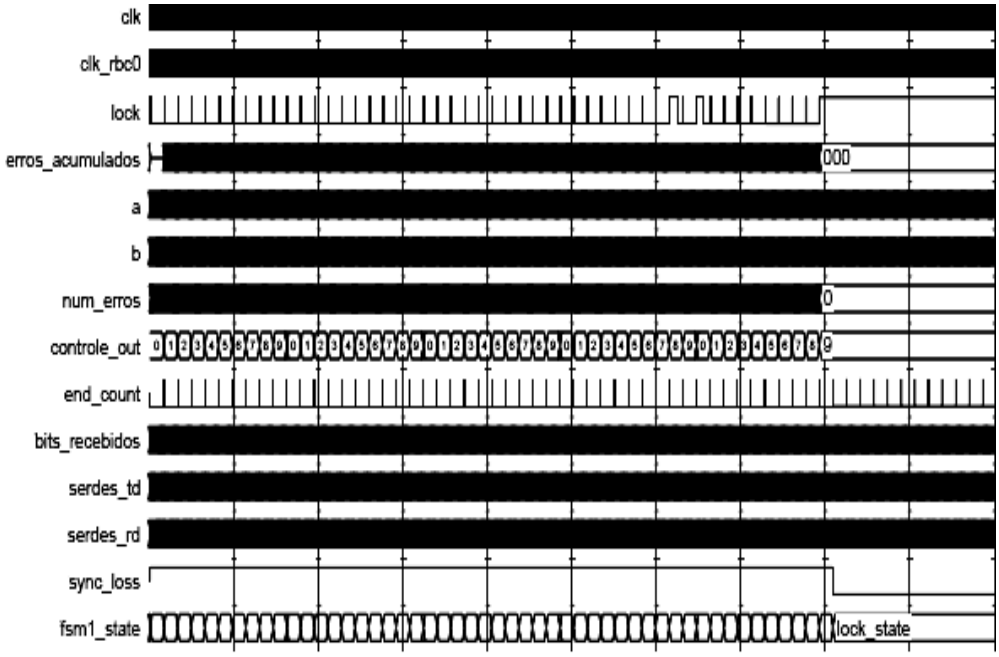

Figura 31. Simulação completa minimizada desde o início até o momento em que o sistema alinha.

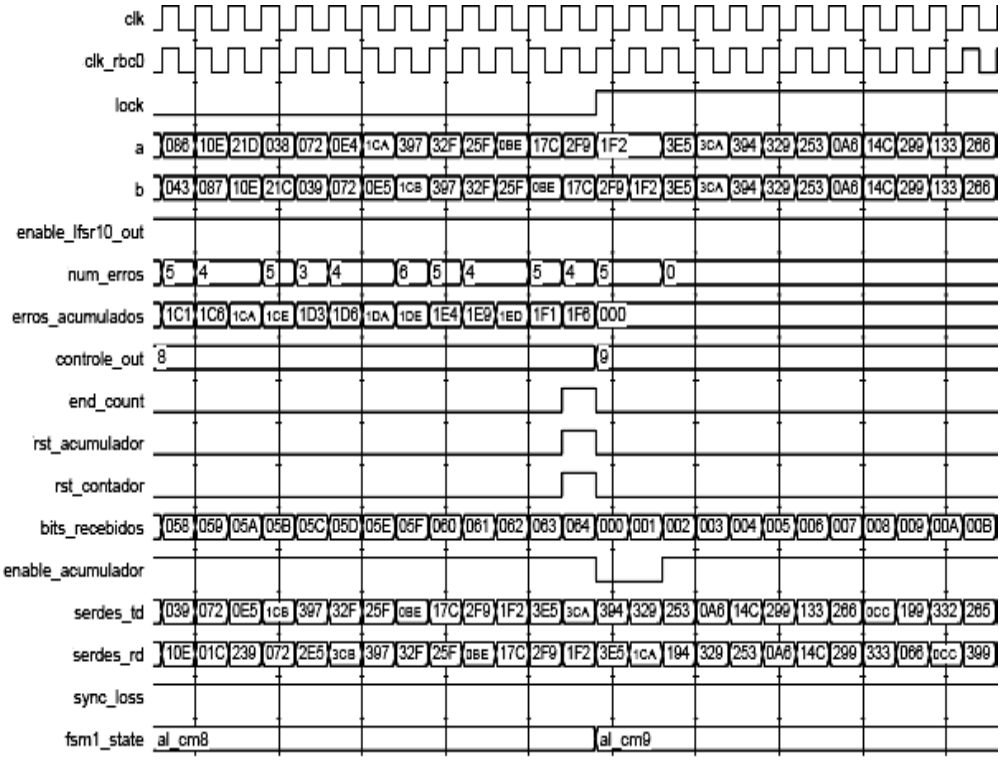

Figura 32. Simulação ampliada do momento em que os erros acumulados do sistema zeraram indicando o alinhamento correto.

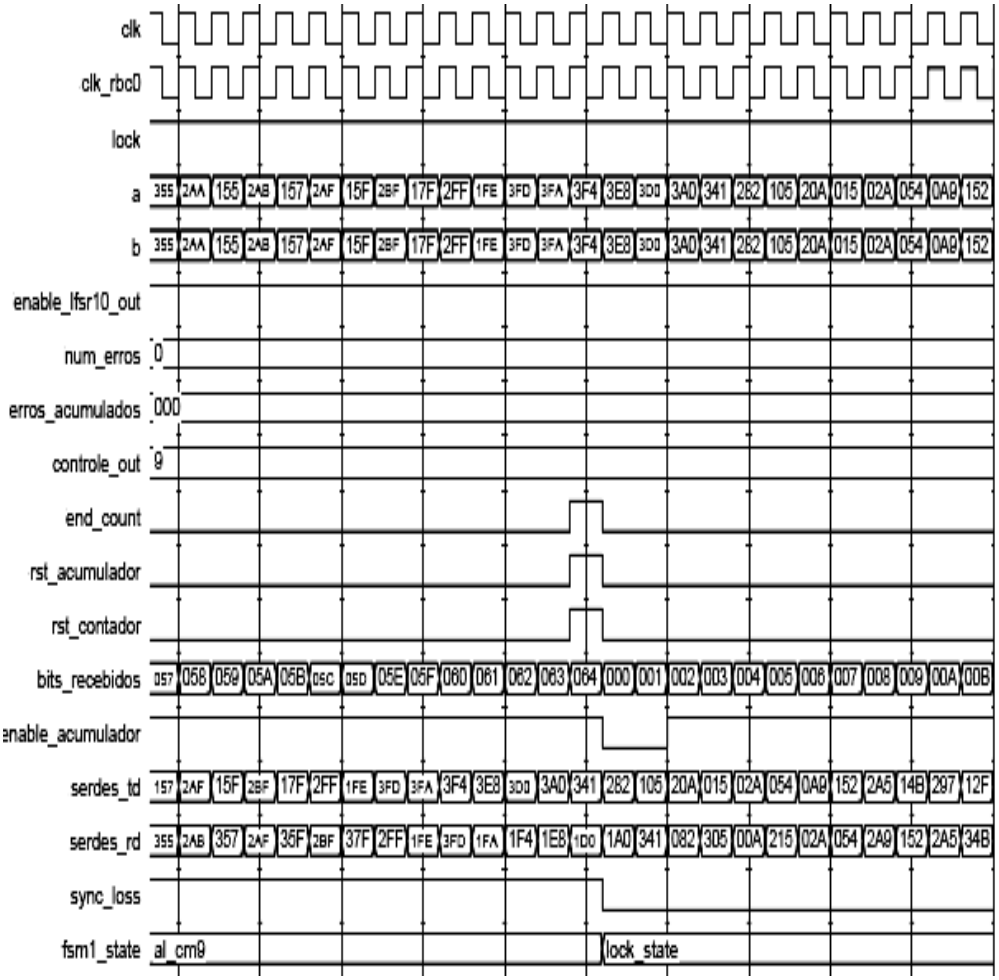

Figura 33. Simulação indicando a transição do 5º estado `alinhamento_comma 9` (`al_cm9`), ao qual o sistema alinhou corretamente, para o estado lock_state de travamento.

5.3.
Análise lógica dos sinais internos ao FPGA

Para analisar os sinais internos ao FPGA utilizou-se o software ChipScope Pro 8.2i da Xilinx. Através desta ferramenta, é possível monitorar com o sistema em funcionamento, algumas interfaces e sinais presentes internamente ao FPGA. Sendo assim, foram selecionados os sinais *SERDES_RD* (10 bits), *dados_alinhados* (10 bits), *erros_acumulados* (10 bits), *bits_recebidos* (10 bits) e *SYNC_LOSS* (1 bit) do bloco AVALIA_ERRO (módulo receptor) para serem analisados. Dessa forma, utilizou-se primeiramente o software ISE Project

Navigator 8.2 e um cabo paralelo JTAG da Xilinx para transferir a programação para a placa. No processo de programação, considerou-se o sinal LOOPEN[11] do chip SERDES sendo igual a zero. Conseqüentemente, utilizou-se conectores externos (loopback externo) entre as saídas diferenciais TXP e TXN, e as entradas diferenciais RXP e RXN para podermos trigar o sistema e analisar o comportamento lógico dos sinais desejados através do ChipScope.

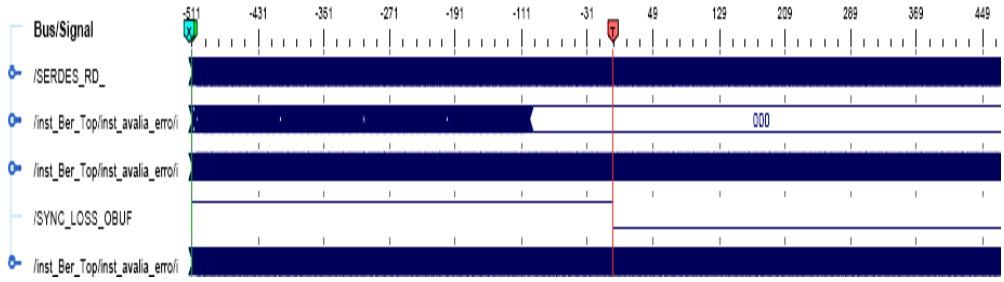

Figura 34. Análise lógica completa e minimizada dos sinais internos ao FPGA através do loopback elétrico.

Na análise acima, trigou-se o sistema quando o sinal *SYNC_LOSS* fosse igual a zero, mas poderíamos trigá-lo através da escolha de outro sinal com outra condição, como veremos mais adiante neste capítulo. Temos em ordem, o sinal *SERDES_RD*, *erros_acumulados*, *bits_recebidos*, *SYNC_LOSS* e *dados_alinhados*.

Pode-se observar que a partir do momento em que o sinal erros_acumulados torna-se nulo, estamos no último estado `alinhamento_comma`, que é responsável pelo alinhamento correto do sistema e após percorrer as 100 palavras de 10 bits deste último estado, há a transição para o estado `lock_state` onde o

[11] A porta de entrada LOOPEN (Loop Enable) do chip SERDES quando habilitada, ativa o loop interno entre a transmissor de dados serias (TD – Transmit Data) e o receptor (RD –Receive Data) do mesmo. Ao desabilitarmos esta porta, temos que realizar o loopback através de conectores externos interligando as portas TXP e TXN (Saída diferencial de transmissão) com as portas RXP e RXN (Entrada diferencial de recepção) sendo que a saída TXP é conectada a entrada RXP e a saída TXN é conectada a entrada RXN. Para trigar o sistema e analisar as interfaces internas ao FPGA utilizou-se o loopback externo tanto elétrico quanto óptico.

sinal *SYNC_LOSS* passa a obter o valor 0 indicando que não há perda de sincronismo.

Ampliando a figura acima podemos analisar melhor as seqüências de entrada e saída do bloco ALINHAMENTO_COMMA. Dessa forma, obtemos a Figura 35 onde as seqüências que chegam em *SERDES_RD* estão desalinhadas e as que saem na porta *dados_alinhados* encontram-se alinhadas. Portanto, pode-se afirmar que o mecanismo de alinhamento proposto no capítulo 4 alinhou corretamente as seqüências garantindo o funcionamento do sistema.

Figura 35. Comparação entre as seqüências de 10 bits do sinal *SERDES_RD* de entrada do bloco ALINHAMENTO_COMMA com as seqüências de 10 bits do sinal

dados_alinhados de saída desse mesmo bloco. Pode-se observar que as seqüências chegam desalinhadas na porta *SERDES_RD* e saem alinhadas na porta *dados_alinhados* através do mecanismo de alinhamento proposto no capítulo 4.

Depois da análise realizada com a montagem acima em loopback elétrico, utilizamos a placa óptica com fibras multimodo e atenuadores no intuito de realizar um loopback óptico, com a condição de somente trigar o sistema caso fosse detectado a presença de algum erro. Primeiramente foram analisados testes sem atenuar a fibra. Em seguida foram introduzidos atenuadores de 5 dB, 10 dB, 15 dB e 20 dB entre as fibras e para cada teste monitorado não foi verificada a presença de nenhum erro.

Este resultado se justifica porque a medição está sendo feita considerando-se 100 palavras de 10 bits, isto é, 1000 bits. Para taxa de erro elevada em uma configuração óptica, por exemplo, BER = 10^{-6}, um bit errado ocorreria para 1 milhão de bits recebidos. Em etapas futuras, quando os erros forem totalizados estes resultados poderão ser verificados.

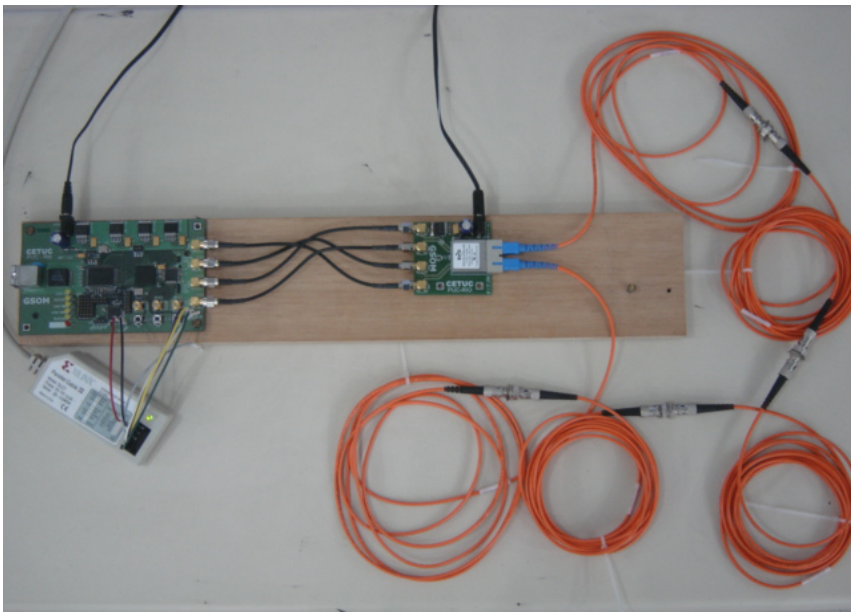

Figura 36. Montagem em loopback óptico onde pode-se monitorar através do ChipScope os sinais internos ao FPGA. Neste software, considerou-se a condição de trigar o sistema somente quando o sinal *erros_acumulados* fosse diferente de zero. Mesmo com

a inserção de atenuadores de 5dB entre as fibras, o sistema não trigou indicando a ausência de erros no teste realizado.

5.4. Conclusão

Ao longo deste capítulo detalharam-se as simulações realizadas entre os módulos transmissor e o receptor. Em seguida, com o sistema em funcionamento, utilizou-se uma ferramenta de monitoração capaz de analisar logicamente os sinais internos ao FPGA considerando-se montagens em loopback elétrico e loopback óptico. Comparando estes resultados obtidos através da análise das interfaces internas do chip com as simulações realizadas em VHDL pode-se garantir o funcionamento correto do sistema e das funcionalidades implementadas no FPGA.