

Proyecto final

Diseño de un bloque de PCS tipo 1000BASE-X

1. Diseñar un bloque de la subcapa de codificación física (PCS, por sus siglas en inglés), de acuerdo con las especificaciones de la cláusula 36 del estándar IEEE 802.3

Como mínimo se debe incluir los bloques señalados en rojo sobre la figura 36-2 (tomada del estándar 802.3)

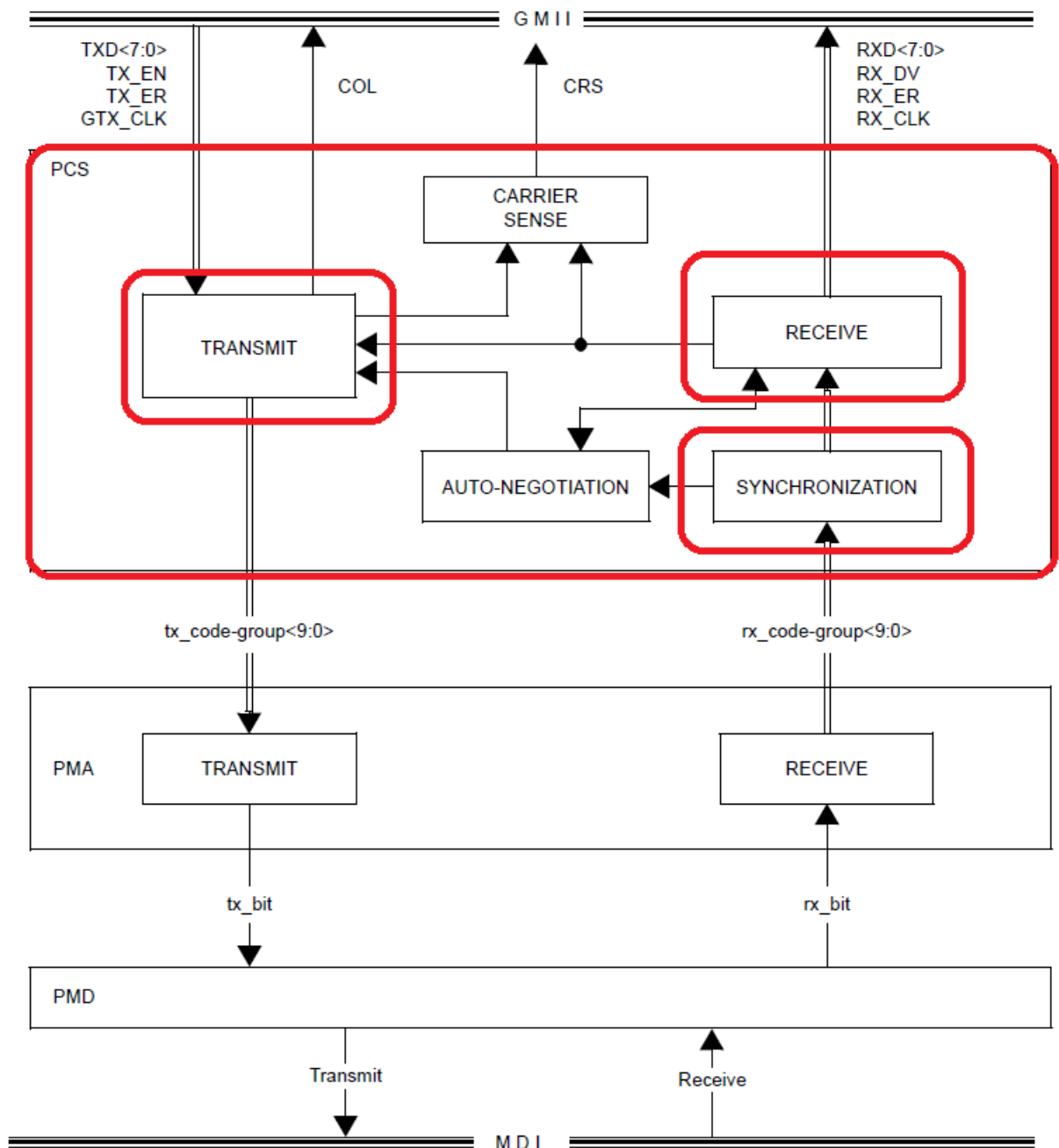


Figure 36-2—Functional block diagram

2. Los demás bloques dentro del PCS (Carrier Sense y Auto-negotiation) se dejan como opcionales. En caso de NO implementarlos, se debe asegurar que las señales que comunican estos bloques con los bloques implementados tengan valores apropiados para el correcto funcionamiento del camino principal (main datapath), tanto en la dirección de transmisión como en la de recepción.
3. El bloque de PCS diseñado debe contar con las interfaces (entradas y salidas) que se muestran en la figura 36-2 y las máquinas de estado deben responder a los diagramas de estados de la sección 36.2.5.2. Para efectos del banco de pruebas, se debe conectar las salidas *tx_code_group* a las entradas *rx_code_group*, de modo que el bloque de PCS quede configurado en modo de *loopback*. Es decir, que los datos que se transmiten por el camino de TX vuelvan a pasar por el PCS en la dirección opuesta, el camino de RX.
4. Como prueba mínima del funcionamiento correcto del bloque de PCS, se espera una prueba de *loopback* que envíe una trama completa de Ethernet, según el diagrama de tiempos de la figura 36-4:

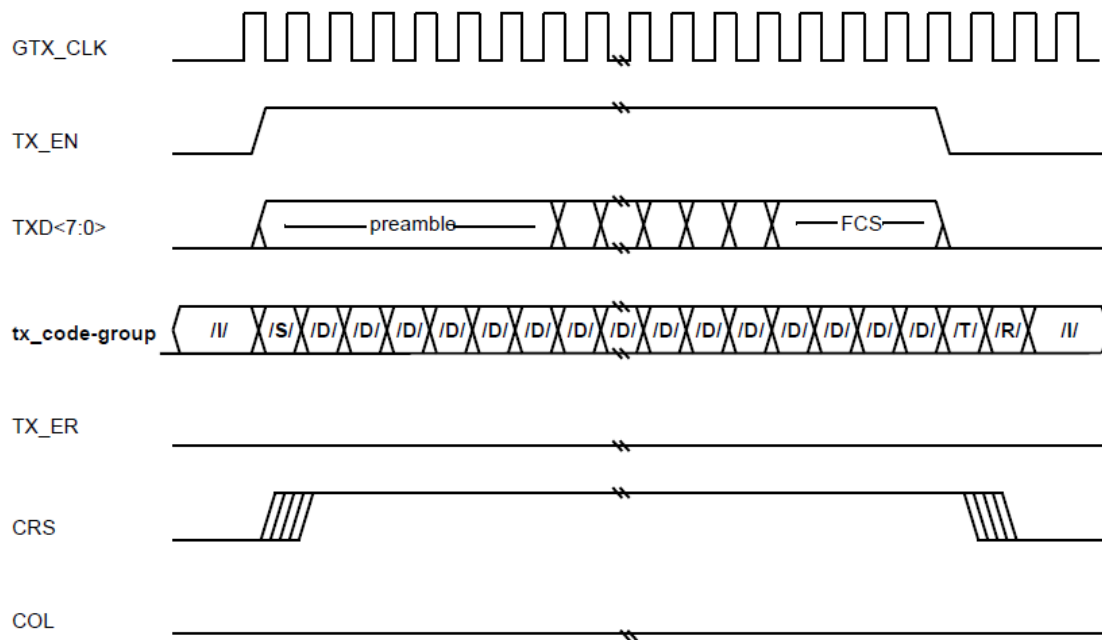


Figure 36-4—PCS encapsulation

5. El dato enviado por la interfaz TXD debe coincidir con el que se recibe en la interfaz RXD al final de la prueba de *loopback*. Es decir, la capa de PCS no debe corromper los datos en ninguno de sus procesos de sincronización o traducción.
6. Cada grupo de trabajo, compuesto por un máximo de 4 integrantes, debe escribir una descripción conductual del PCS usando Verilog. Esta descripción servirá como una especificación detallada y formal del funcionamiento del dispositivo diseñado.
7. La descripción en Verilog deberá tener al menos un módulo de banco de pruebas, un módulo probador, un módulo de “envoltorio” (*wrapper*) que contenga a todos los submódulos del PCS y al menos un submódulo para cada máquina de estados . Puede

usar EDA playground para desarrollar el proyecto.

8. Definir un plan de pruebas para garantizar el funcionamiento del diseño. El plan de pruebas debe incluir, como mínimo, la prueba de loopback especificada en el punto #4. El módulo probador debe suministrar las señales necesarias para que las pruebas se realicen.
9. Con el fin de facilitar el proceso de revisión, se le solicita organizar los entregables del proyecto de la siguiente manera:
 - i. Reporte en formato PDF cuyo nombre debe seguir el patrón Grupo_<# de grupo>.pdf, por ejemplo, Grupo_1.pdf y **que deberá ser entregado en la carpeta correspondiente del sitio del curso en Mediación Virtual.**
 - ii. Uno o varios archivos de Verilog con el formato <nombre de archivo>.v que construyan la solución que se solicita en el proyecto, incluyendo probadores y bancos de pruebas, **que deberán ser entregados en el repositorio de GitHub asignado por el profesor para cada grupo.**
 - iii. Un archivo Makefile que permita correr todos los pasos de simulación con una sola línea de comando, tal como se ha venido haciendo en las tareas. **Este archivo también debe subirse a Github**, junto con el código de Verilog.
- b. El banco de pruebas, testbench.v, deberá incluir a todos los demás archivos *.v, de modo que la compilación y simulación del testbench implique la compilación y simulación de todos los archivos internos.
- c. El probador, tester.v, debe escribirse de forma tal que una sola simulación contenga los resultados de la misma prueba de loopback que se describe en el reporte.
- d. El Makefile debe contener, como mínimo, los comandos necesarios para correr la compilación y simulación de los módulos del proyecto, tal como se ha manejado en tareas anteriores.

Calificación:

El proyecto final representa 40% de la nota del curso y se califica en cinco rubros separados. A pesar de que el proyecto es un trabajo grupal, para cada estudiante, la mitad de la calificación del proyecto corresponde a rubros individuales y la otra mitad a rubros grupales, como se muestra a continuación:

1. **Avance de resultados (rubro grupal):** Representa 10% de la nota del curso. Corresponde a un informe escrito describiendo el avance del proyecto final hasta el momento. Tareas realizadas, tareas pendientes, resultados hasta ahora, problemas encontrados, observaciones relevantes, etc. **El informe escrito se debe entregar en la página de Mediación Virtual en la carpeta disponible para tal fin.**
2. **Informe final (rubro grupal):** Representa 10% de la nota del curso. Corresponde a un informe escrito describiendo el desarrollo del proyecto completo y los resultados obtenidos, de forma similar a como se hizo en las tareas individuales, pero sin la restricción de 10 páginas como máximo. **El informe escrito se debe entregar en la página de Mediación Virtual en la carpeta disponible para tal fin.**
3. **Calificación del código en GitHub (rubro individual):** Representa un 10% de la nota del curso y corresponde a la evaluación del profesor de los aportes de cada estudiante al repositorio particular de su grupo de trabajo.
4. **Presentación oral (rubro individual):** Representa 5% de la nota del curso. Tanto el avance de resultados como el informe final deberán ser presentados de forma oral por los miembros del grupo. Para cada una de esas dos presentaciones, cada grupo dispondrá de una ventana de 10 minutos para exponer y 5 minutos adicionales para comentar preguntas y observaciones. En cada presentación, cada grupo deberá designar dos voceros diferentes, de modo que al final del semestre las cuatro personas que integran el grupo hayan tenido la oportunidad de exponer. La exposición oral y la respuesta a las preguntas de estas sesiones se calificará de forma individual.
5. **Calificación de compañeros (rubro individual):** Representa 5% de la nota del curso y corresponde al promedio de la calificación otorgada por el resto de integrantes de su grupo de trabajo. El profesor enviará un formulario a cada estudiante para la calificación de sus compañeros.

Guía para el reporte

Se debe entregar en forma electrónica un documento que incluya los siguientes puntos:

1. **Resumen:** Breve (Media página máximo) descripción de todo el proyecto. Esta sección es fundamental pues puede determinar si el lector se interesa o no en leer los detalles del proyecto. Un resumen mal hecho puede esconder un excelente proyecto. El resumen debería incluir:
 - a) Descripción breve del sistema, es decir, qué hace. Incluya alguna característica que considere que distingue este diseño en particular.
 - b) Las pruebas que se realizaron y qué resultados se obtuvieron. Indique problemas que se tuvieron que considere importante resaltar.
 - c) Conclusiones más importantes y recomendaciones para un diseño posterior.
2. **Descripción Arquitectónica:** Incluye un diagrama de bloques con las señales más importantes que sirve como base para describir el funcionamiento del sistema. La descripción va en términos de lo que se espera que el sistema haga. Es decir, se debe detallar la funcionalidad del sistema, el protocolo de las señales que se usan para que funcionen cada una de las partes y las secuencias de eventos que se deben dar. Esta descripción podría ir acompañada de tablas de verdad, tablas de transición de estados, diagramas de estados, diagramas temporales, etc.
3. **Plan de Pruebas:** Aquí se deben enumerar, esto es, se debe presentar una **lista detallada** de las pruebas que se le van a hacer al diseño para verificar que está funcionando de acuerdo a las especificaciones dadas. La lista debe contener por lo menos los siguientes elementos i) Nombre/número de prueba, ii) Descripción de la prueba, y iii) Una indicación de si el diseño la falló o la pasó. Estas pruebas podrían incluir la generación de vectores de entrada para probar en forma exhaustiva todas las líneas de una tabla de verdad o tabla de estados, patrones aleatorios de entradas para tratar de causar errores en la respuesta del diseño, o patrones específicos que ejerciten un cierto modo de funcionamiento. Cada prueba debería ser claramente enumerada en el plan para que también se pueda hacer referencia a ella en el código del banco de pruebas del diseño.
4. **Instrucciones de utilización de la simulación:** Esta sección debe mostrar los comandos necesarios para hacer funcionar la simulación en todos los casos que especifica el plan de pruebas. Hay que suponer que el diseño de un grupo puede ser utilizado por otro grupo o el profesor. Si los resultados no se pueden repetir porque no se conocen los comandos para hacer funcionar la simulación entonces es como si el diseño no funcionara del todo. Se recomienda crear un Makefile de modo que se pueda correr todas las pruebas del caso con un solo comando en Icarus Verilog y GTKwave.
5. **Ejemplos de los resultados:** Una descripción de los resultados más importantes acompañados de los diagramas temporales de la simulación o cualquier otra salida que demuestre claramente el comportamiento descrito. No es necesario incluir una muestra exhaustiva de resultados, sino que los más representativos del diseño. El punto es mostrarle al lector los comportamientos más sobresalientes para formarle una idea clara del funcionamiento del diseño. Ya verá el lector si desea más detalles, entonces podrá correr una simulación.

6. **Conclusiones y recomendaciones:** Basado en los resultados obtenidos se indica aquí qué se logró con el proyecto. Puede ser que se concluya que con el diseño propuesto se tiene una limitación en la velocidad de respuesta de... etc. O que con ciertas combinaciones de entradas el diseño se vuelve inestable o los resultados no son los esperados. También se puede concluir qué ventajas o problemas encontraron al seguir el plan de trabajo. A raíz de las conclusiones se puede también recomendar cómo se podría mejorar el diseño o qué otras pruebas se le podrían hacer para garantizar su funcionamiento en otras condiciones que al principio no se consideraron, o también cómo se debería planear el siguiente proyecto para poder cumplirlo a tiempo.