

Plan de Verificación (Testplan)

Barrantes Sebastian - Ruiz Tayron

Verificación funcional de circuitos integrados/EL5811

Profesor: Dr.-Ing. Ronny García Ramírez

Índice

1. Set de pruebas funcionales	2
1.1. Pruebas funcionales básicas	2
1.2. Pruebas de estrés	2
1.3. Pruebas de error	2
1.4. Pruebas de interrupciones	3
1.5. Pruebas de configuración (APB)	3

1. Set de pruebas funcionales

1.1 Pruebas funcionales básicas

- **Objetivo:** Verificar alineación correcta para combinaciones válidas de CTRL.SIZE y CTRL.OFFSET.
- **Estímulos aleatorios:** num_txns en rango[50..100]; size $\in \{1,2,4, W/8\}$; offset $\in [0,1,2,3]$ restringido a $((W/8)+offset) \% size == 0$; tiempo entre transacciones en rango[0..5] ciclos.
- **Checks:** Scoreboard de referencia (corrimiento por bytes) debe igualar md_tx_{data,offset,size} no debe subir md_rx_err; CNT_DROP permanece constante; pslverr==0.
- **Cobertura:** Cruzar legal SIZE con legal OFFSET; cobertura de primeras y últimas palabras de ráfaga.
- **Métricas :** tasa de alineación correcta, latencia media TX; throughput (palabras/ciclo).

1.2 Pruebas de estrés

- **Objetivo:** Forzar límites de FIFO y temporización.
- **Estímulos aleatorios:** ráfagas largas num_txns en rango[500..1000]; tiempo entre transacciones (picos y valles): brechas con alta actividad en rango[0-1]ciclos y brechas con baja actividad[50..100] ciclos; md_tx_ready con pausas en 0 durante tiempos en rango [10..100] ciclos. Tamaños con sesgo a máximos definidos en size=4 y offset = 0.
- **Checks:** STATUS.{RX_LVL, TX_LVL} alcanzan niveles de 0, 1, MAX-1, MAX; no hay pérdidas ni duplicados de datos; IRQs de FULL/EMPTY se setean y se limpian correctamente; no hay deadlocks.
- **Cobertura:** bins para niveles FIFO críticos; verificar la correcta distribución de tiempos.
- **Métricas:** utilización promedio de FIFO; throughput sostenido; tiempo de recuperación tras pausa en md_tx_ready; cero desalineaciones.

1.3 Pruebas de error

- **Objetivo:** Validar manejo de errores de RX y contador de drops.

- **Estímulos aleatorios:** inyectar proporción $p_{\text{illegal}} \in [50\%]$ de transacciones con `size==0` o `((W/8)+offset)%size != 0`; número de transacciones = 600.
- **Checks:** cada transacción ilegal aceptada (`valid&ready`) produce `md_rx_err=1`, no genera salida en TX, e incrementa `STATUS.CNT_DROP` con saturación a MAX; transacciones legales siguen correctas; IRQ de MAX_DROP solo cuando corresponde; verificar la funcionalidad de CLR del registro de control.
- **Cobertura:** tipos de ilegalidad con `size=0` vs desalineado ilegal; saturación de `CNT_DROP` en $\{MAX-1 \rightarrow MAX\}$; secuencias legales e ilegales mezcladas.
- **Métricas:** porcentaje de errores detectados tiene que ser igual a 100% teniendo en cuenta que se calcula como los `errores_reportados` entre `errores_inyectados`; tasa de falsos positivos tiene que ser 0; tiempo de propagación de `md_rx_err`.

1.4 Pruebas de interrupciones

- **Objetivo:** Verificar procesos IRQ y registros `IRQEN` y `IRQ`.
- **Estímulos aleatorios:** habilitar aleatoriamente subconjuntos de IRQs; generar escenarios que detonen `RX_EMPTY/FULL`, `TX_EMPTY/FULL`, `MAX_DROP`; alternar `md_tx_ready` para forzar `EMPTY/FULL`.
- **Checks:** bits en `IRQ` se ponen solo cuando el evento ocurre y `IRQEN` lo permite; lectura/limpieza `W1C` funciona, 1 limpia y 0 preserva; `irq` refleja OR de bits activos; los bits de estado del `IRQ` no se limpien solos ni por causas ajenas a una acción válida del software.
- **Cobertura:** Para cada fuente de IRQ se va a estimular `IRQEN={0,1}` y `W1C={limpia, no limpia}`; coincidencia de múltiples IRQs simultáneos.
- **Métricas:** latencia IRQ; tasa de limpieza exitosa; ausencia de IRQ fantasma.

1.5 Pruebas de configuración (APB)

- **Objetivo:** Robustez de mapa de registros y políticas de error.
- **Estímulos aleatorios:** escrituras y lecturas a `CTRL/STATUS` con orden aleatorio; cambios de `CTRL.SIZE` y `CTRL.OFFSET` durante tráfico; accesos ilegales como direcciones fuera de mapa, escribir `STATUS`, combinaciones inválidas.
- **Checks:** `pready` correcto; `pslverr=1` en accesos ilegales; `CTRL` altera comportamiento en la siguiente transacción con reglas claras; `STATUS` refleja niveles de FIFO y contador DROP.

- **Cobertura:** todas las direcciones válidas; corroborar patrones de acceso inválido; cambios de CTRL en distintas fases de actividad RX/TX.
- **Métricas:** tasa de APB errores detectados sobre inyectados; tiempo de aplicación efectiva de nueva config (ciclos); consistencia de registro STATUS y FIFOs/CNT DROP