

Universidad Nacional de La Matanza

Unidad Académica: Ingeniería

Código: C163

Título del Proyecto: Evaluación de la Confiabilidad de FPGAs

Director del Proyecto: Dr. Ing. José Luís Roca

Integrantes: Ing. Ricardo Oscar Juliá

Ing. Ariel Serra

Fecha de inicio: 01/01/2014

Fecha de finalización: 31/12/2015

Resumen:

Los beneficios que ha promovido el uso de las así denominadas FPGAs (del ingles “Field Programmable [Gate Arrays](#)” ha dado lugar a un amplio espectro de utilización que va desde productos de consumo masivo a astronáutica. Esta diversidad exige de algún modo la necesidad de evaluar su comportamiento a lo largo del tiempo, esto es su confiabilidad, principalmente debido a su susceptibilidad a presentar errores de software o programación, debido a la alta densidad de celdas embebidas. La evaluación de la confiabilidad es un paso importante a la hora de diseñar sistemas altamente confiables, que se traduce en una fuerte ventaja competitiva en el mercado actual. Es así que el análisis de confiabilidad de circuitos digitales se esta convirtiendo en una característica importante en el proceso de diseño de sistemas de escala manométrica. Comprender las relaciones entre la estructura del circuito y su confiabilidad permite al diseñador implementar y de hecho, tomar alguna decisión que mejore el mismo diseño preliminar.. El presente trabajo presenta un modelo probabilístico sobre estructuras lógicas propias de las FPGAs, esto es sobre los distintos CLB (Complex Logic Blocks), usualmente constituidos LUTs (Look-up Tables) y Biestables (Flip-Flop) interconectados vía PSMs (Programmable Switch Matrix) y IOBs (Input-output Blocks). Tomando como base de evaluación, los modelos de circuitos lógicos propuestos en varias publicaciones (Reliability Evaluation of Logic Circuits”, Microelectronics & Reliability, Vol. 25, Nº 2, pp.257-260, Pergamon Press Ltd.) por el director del presente proyecto y ampliando estos conceptos a nivel de sistema, se propone investigar y analizar la posibilidad de utilizar esta metodología en la evaluación de la confiabilidad de las estructuras emergentes de la utilización de FPGAs en desarrollos electrónicos. Asimismo poner como objetivo analizar y estudiar la interrelación entre lenguajes de programación HDL (Hardware Description Language) y las estructuras emergentes de la modelización propuesta.

Palabras Claves: FPGAs (Field Programmable Gate Arrays), Confiabilidad, CLB (Complex Logic Blocks), LUTs (Look-up Tables), PSMs (Programmable Switch Matrix), IOBs (Input-output Blocks), SMRA (Static Random Access Memory), CBs (Connection Blocks).

Área de Conocimiento: Ingeniería de Comunicaciones, Electrónica y Control

Código de Área de Conocimiento: 1800

Disciplina: Computación

Código de Disciplina: 1802

Campo de Aplicación: Ciencia y Tecnología

Código de Campo de Aplicación: 1040

Otras dependencias de la UNLaM que intervinieron en el proyecto: Ninguna

Otras instituciones intervinientes en el Proyecto: CONAE

Otros proyectos con los que se relaciona: Ninguno

Título del Proyecto: Evaluación de la Confiabilidad de FPGAs

RESUMEN

Los beneficios que ha promovido el uso de las así denominadas FPGAs (del inglés "Field Programmable Gate Arrays" ha dado lugar a un amplio espectro de utilización que va desde productos de consumo masivo a astronáutica. Esta diversidad exige de algún modo la necesidad de evaluar su comportamiento a lo largo del tiempo, esto es su confiabilidad, principalmente debido a su susceptibilidad a presentar errores de software o programación, debido a la alta densidad de celdas embebidas. La evaluación de la confiabilidad es un paso importante a la hora de diseñar sistemas altamente confiables, que se traduce en una fuerte ventaja competitiva en el mercado actual. Es así que el análisis de confiabilidad de circuitos digitales se esta convirtiendo en una característica importante en el proceso de diseño de sistemas de escala manométrica. Comprender las relaciones entre la estructura del circuito y su confiabilidad permite al diseñador implementar y de hecho, tomar alguna decisión que mejore el mismo diseño preliminar.. El presente trabajo presenta un modelo probabilístico sobre estructuras lógicas propias de las FPGAs, esto es sobre los distintos CLBs (Complex Logic Blocks), usualmente constituidos LUTs (Look-up Tables) y Biestables (Flip-Flop) interconectados vía SWs (Programmable Switch Matrix) y IOBs (Input-output Blocks). Tomando como base de evaluación, los modelos de circuitos lógicos propuestos en varias publicaciones ("Reliability Evaluation of Logic Circuits", Microelectronics & Reliability, Vol. 25, Nº 2, pp.257-260, Pergamon Press Ltd.) y ampliando estos conceptos a nivel de sistema, se propone investigar y analizar la posibilidad de utilizar esta metodología en la evaluación de la confiabilidad de las estructuras emergentes de la utilización de FPGAs en desarrollos electrónicos. Asimismo poner como objetivo analizar y estudiar la interrelación entre lenguajes de programación HDL (Hardware Description Language) y las estructuras emergentes de la modelización propuesta. A esos efectos se utilizara el software RELEX de propiedad de la Comisión Nacional de Actividades Espaciales CONAE para modelar el sistema, trasladando la investigación y análisis al plano del Fault Tree Análisis (FTA).

Cabe mencionar que no se realizaron publicaciones hasta el momento. El trabajo de investigación es demasiado extenso para ser publicado tal cual está en el informe presentado. Es de por sí, para el que lo lee extensivamente denso y complejo. Debido a lo expuesto precedentemente es necesario preparar una publicación más compacta. Se está trabajando en compactar todo el trabajo de investigación a esos fines.

1. Hipótesis de trabajo.

Se tomara como hipótesis de trabajo un entorno de aplicación de una FPGA que resulte conveniente para bajar la cantidad de restricciones impuestas en un uso particular. Esto conjuntamente con el "Heritage" actual en cuanto a diversidad de fabricantes generará resultados esperados más cercanos a escenarios estándares. Así mismo se tomaran como puntos de partida los modelos de circuitos lógicos propuestos en varias publicaciones (Reliability Evaluation of Logic Circuits", Microelectronics & Reliability, Vol. 25, Nº 2, pp.257-260, Pergamon Press Ltd.). El programa utilizado para transvasar la modelización en árboles de falla será el programa Relex de propiedad de la Comisión Nacional de Actividades Espaciales CONAE.

2. Estado actual de conocimiento.

Mucho es lo que hay desarrollado sobre FPGAs desde el punto de vista hardware y en cuanto a evaluación de su confiabilidad fijando la atención en el ambiente de trabajo de las mismas, sobre todo en el área espacial. Mucha información proviene de los distintos modos de falla acaecidos en vuelo, dando origen a información sobre fallas del tipo fijas y otras transitorias. Asimismo existe información emergente de tests llevados a cabo en laboratorios especializados e independientes, donde NASA comparte experiencia. La bibliografía es escasa, sin embargo, respecto a proveer estimaciones cuantitativas a partir de la disposición y de modelos matemáticos de un dispositivo tan versátil como una FPGA. En general tampoco es mucha la información respecto a estimación cuantitativa de diseños HDL (Hardware Description Language). El estado actual de conocimiento no provee este tipo de modelizaciones y la generación de un modelo matemático que permita la extracción de valores cuantitativos de la confiabilidad de una FPGA en función de la estructura interna emergente de su programación HDL.

3. Presentación de la problemática a investigar.

Una FPGA es un dispositivo semiconductor que contiene bloques de lógica cuya interconexión y funcionalidad puede ser configurada *'in situ'* mediante un lenguaje de descripción especializado. La lógica programable puede reproducir desde funciones tan sencillas como las llevadas a cabo por una puerta lógica o un sistema combinacional hasta complejos sistemas en un chip. Las FPGAs se utilizan en aplicaciones similares a los ASICs sin embargo son más lentas, tienen un mayor consumo de potencia y no pueden abarcar sistemas tan complejos como ellos. A pesar de esto, las FPGAs tienen las ventajas de ser reprogramables (lo que añade una enorme flexibilidad al flujo de diseño), sus costos de desarrollo y adquisición son mucho menores para pequeñas cantidades de dispositivos y el tiempo de desarrollo es también menor. Siendo estos dispositivos de gran utilidad y uso generalizado en toda la industria desde la electrónica de entretenimiento hasta la astronáutica es que es necesario evaluar su confiabilidad ya sea desde el punto de vista Hardware como Software. La investigación propuesta resultara útil en el sentido de acompañar al desarrollista y programador de modo de poder lograr diseños confiables acordes a la utilización de FPGAs en los ámbitos correspondientes.

4. FPGA genérica como base de análisis.

A los efectos de evaluar la confiabilidad de una FPGA se optó por tomar como genérica una FPGA con memoria estática (SRAM) constituida por 20 CLBs (Complex Logic Blocks). Cada uno de ellos ruteado vía CBs (Conexión Blocks) y SWs (Switch Matrix) hacia los IOBs (Input-output Blocks). El ruteo consiste en 4 vías horizontales y 4 verticales. Las vías pueden interconectarse dentro de cada SW programable. Cada una de estas vías puede conectarse en una de las tres direcciones posibles (seguir el mismo camino o ir hacia la izquierda o a la derecha del mismo) o ir en más de una dirección. Como existen cuatro vías entrando y saliendo de cada SW se requieren de 6 transistores de paso más sus respectivas SRAMs para comandar las líneas en cualquier dirección adecuada. Se utiliza ruteo segmentado, esto implica que no todas las vías son de la misma longitud. Hay segmentos de diversas longitudes. En nuestro caso las longitudes son 1, 2, 3 y 4. Ver Figs.1 y 2. Como se observa si los segmentos están apropiadamente distribuidos existen solamente 8 vías (dos de cada lado de las SWs), reduciendo el número de transistores de paso y SRAMs a 12 por SW. Ver Fig.3. Los transistores de paso utilizan transistores de realimentación para retornar las señales hacia el terminal de alimentación. La arquitectura de ruteo se completa con las CBs. Ver Fig.4 y 5. Estas proveen conexión entre los CLBs ya las vías de ruteo. Dos son los tipos de conexión provistas por las CBs. CB_1 desde las vías de ruteo hasta las entradas de los CLBs y CB_2 desde las salidas de los CLBs hacia las vías de ruteo. Solo una de las vías de ruteo puede conectarse con cualquiera de las entradas de modo de utilizar multiplexores a esos efectos. Las salidas por otro lado puede conectarse a cualquier número de vías de ruteo para lo cual se utilizan transistores de paso que conectan la salida a cualquiera de las vías de ruteo. Los CLBs están constituidos básicamente por un LUT de orden 4. Ver Fig.6.

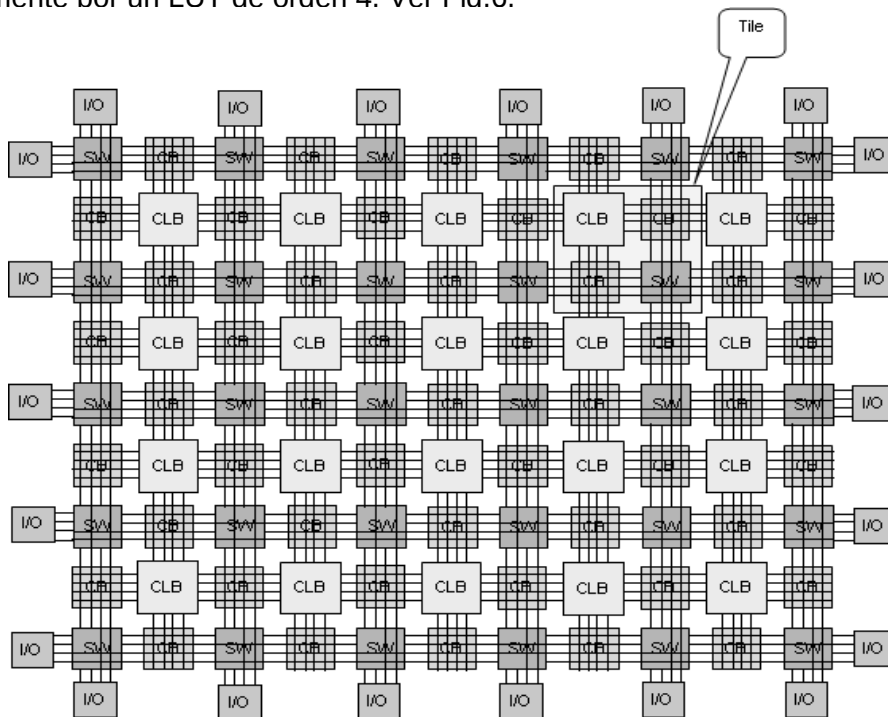
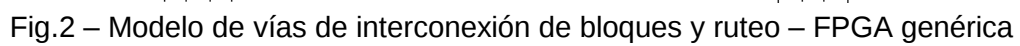


Fig.1 – Modelo de FPGA genérica



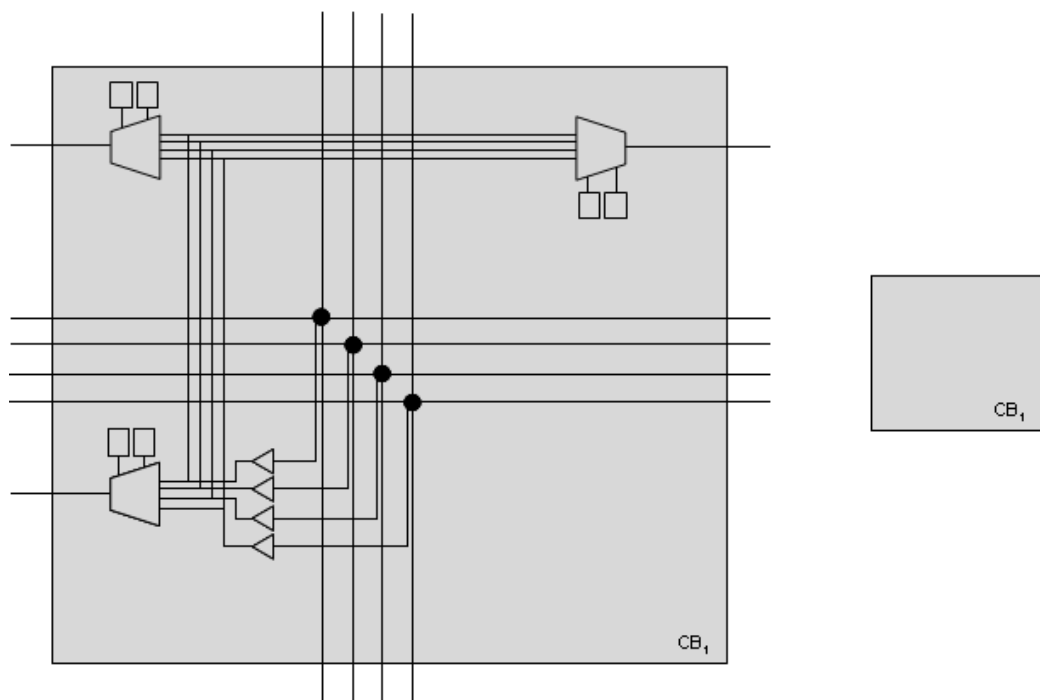


Fig.4 – Modelo del Bloque de interconexión CB_1 – FPGA genérica

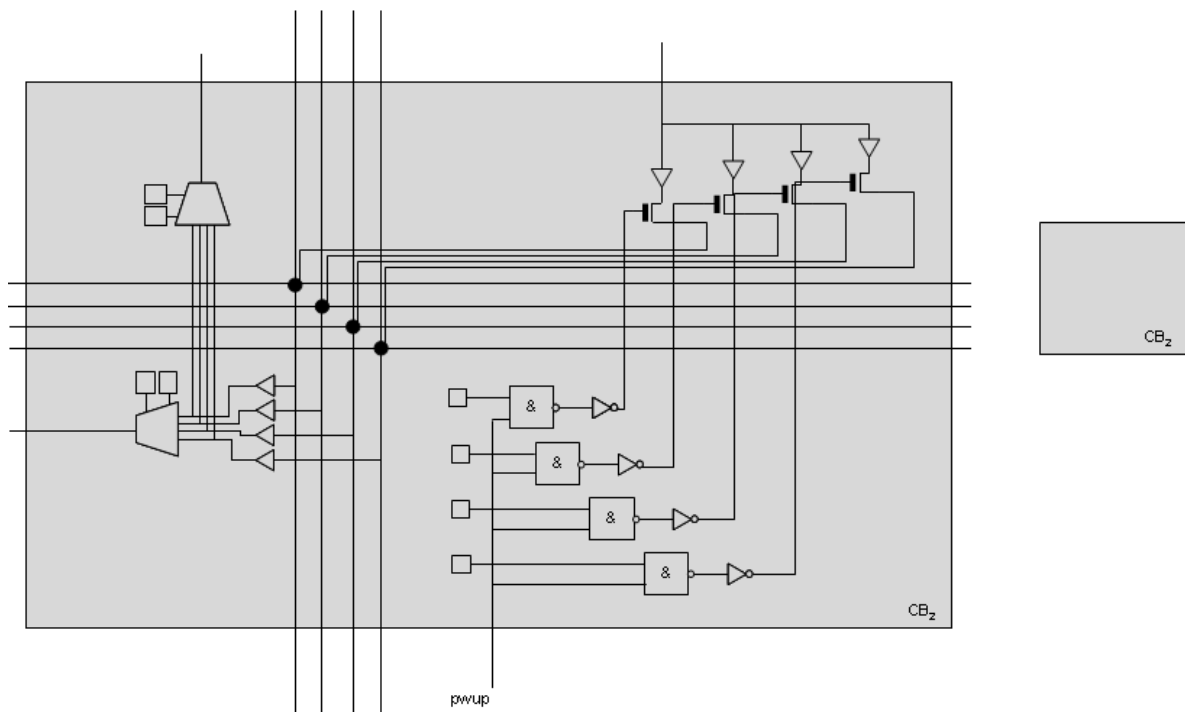


Fig.5 – Modelo del Bloque de interconexión CB_2 – FPGA genérica

Los CLBs

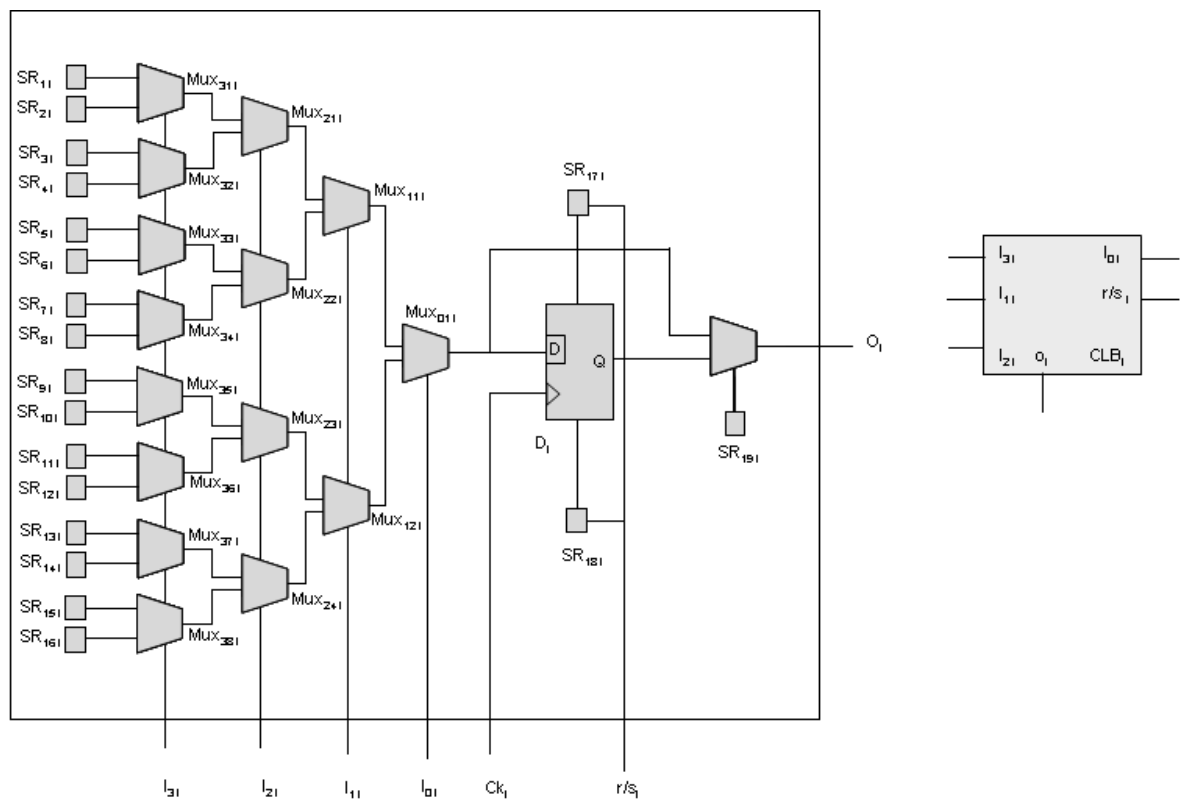


Fig.6 – Modelo de Complex Logic Block – FPGA genérica

La FPGA genérica tendrá 22 bloques de entrada salida IOBs de acuerdo a las vías de ruteo provenientes de las SWs. Ver Fig.7. La programación es efectuada vía SRAM cuyo esquema se observa en la Fig.8

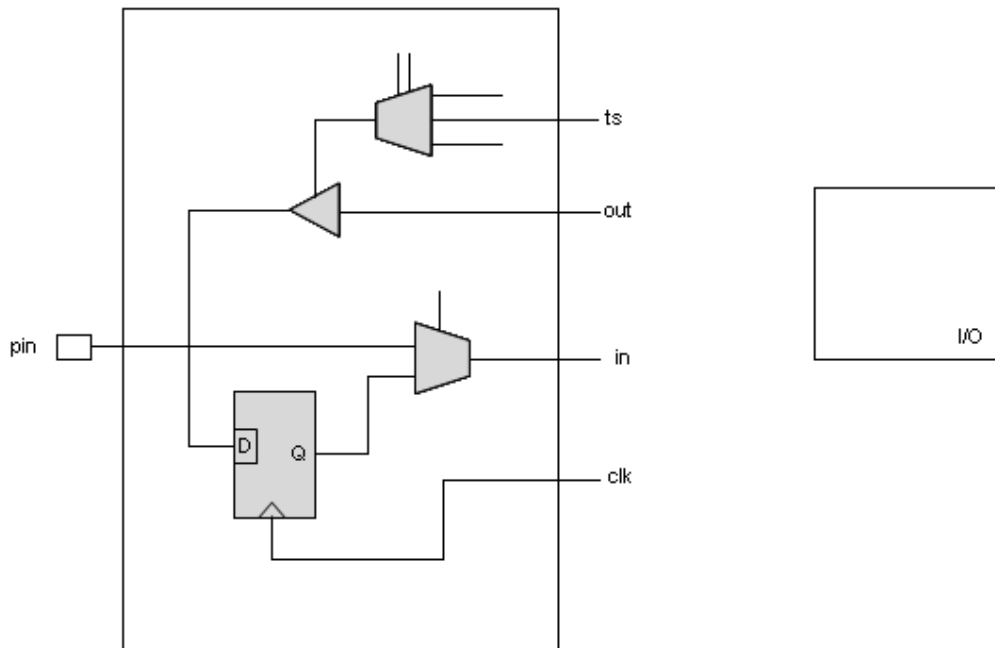


Fig.7 – Modelo de Input – Output Block – FPGA genérica

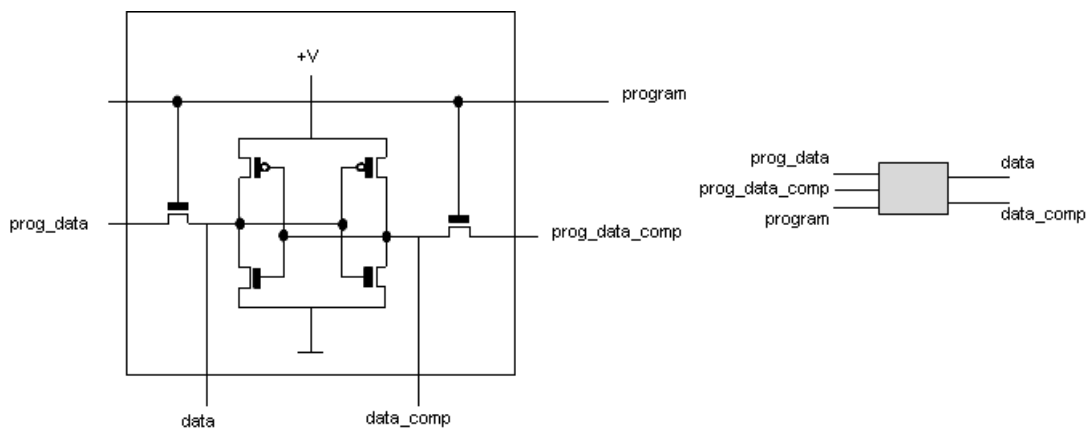


Fig.8 – Modelo de SRAM – FPGA genérica

El número de SRAMs por Tile a esos efectos es de 45. El número de SRAMs puede evaluarse de la siguiente manera:

SRAMs para la CB_1

2 Entradas hacia el CLB por la izquierda

1 para el LUT de orden 4

1 para el SET-RESET

1 Entrada hacia el CLB por la derecha

1 para el LUT de orden 4

Cada entrada requiere 2 SRAMs.

Total SRAMs en CB_1 = 6 SRAMs

SRAMs para la CB_2

1 Entrada hacia el CLB por arriba

1 para el LUT de orden 4

1 Salida desde el CLB por arriba

1 para el LUT de orden 4

1 Entrada hacia el CLB por debajo

1 para el LUT de orden 4

Cada entrada requiere 2 SRAMs.

Total SRAMs en CB_1 = 8 SRAMs

SRAMs para la SW

2 Vías de ruteo hacia cualquiera de las tres direcciones

Cada vía requiere 6 SRAMs

Total SRAMs en CB_1 = 12 SRAMs

SRAMs para el CLB

16 Entradas a los Multiplexores del LUT de orden 4

Cada entrada requiere 1 SRAM

2 Entrada para el SET-RESET

Cada entrada requiere 1 SRAM

1 Entrada para el Multiplexor de Salida del CLB

Cada entrada requiere 1 SRAM

Total SRAMs en CLB = 19 SRAMs

5. Modelos de Falla de Bloques Lógicos genéricos

Los modelos de falla de los circuitos lógicos básicos se observan en las Fig. 9, 10, 11 y 12, 13 y 14 corresponden a una compuerta AND, NAND, OR, NOR; MUX; NOT, Biestable D y SDRAM. El modelo de fallas para el En los modelos presentados se ha trasladado el esquema de errores a la salida de los dispositivos en función de los errores en las entradas y del valor actual de las mismas, además de contemplar la probabilidad de falla propia del dispositivo que figura como W en los modelos. Esto implica como consecuencia fáctica la posibilidad de que los dispositivos digitales puedan trasladar en el circuito y de uno a otro las fallas intrínsecas de los mismos dependiendo en gran medida del valor digital de las variables de entrada al circuito. La modelización sistemática se basa en mi publicación "Reliability Evaluation of Logic Circuits", Microelectronics & Reliability, Vol. 25, N° 2, pp.257-260, Pergamon Press Ltd. El punto crucial fue trasladar estos modelos al lenguaje de árbol de fallas (FTA), para luego mediante el programa RELEX-Reliability Studio. PTC Company USA, propiedad de CONAE, delinear los modelos reales a utilizar en la simulación de fallas. Los modelos básicos planteados servirán como fundamento de los modelos de cada uno de los componentes citados en el párrafo 4.

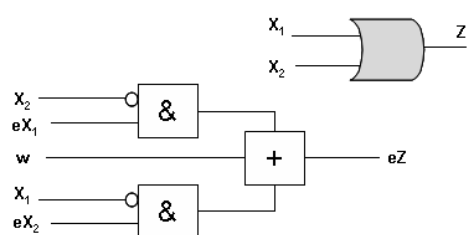


Fig.9 – Modelo de fallas OR

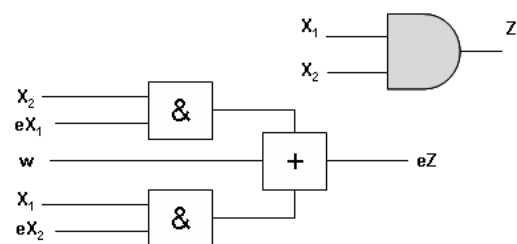


Fig.10 – Modelo de fallas AND

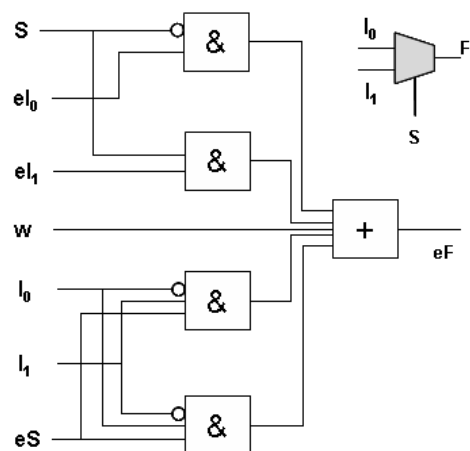


Fig.11 – Modelo de fallas MUX 2 Entradas

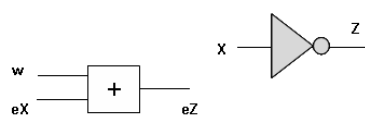


Fig.12 – Modelo de fallas NOT

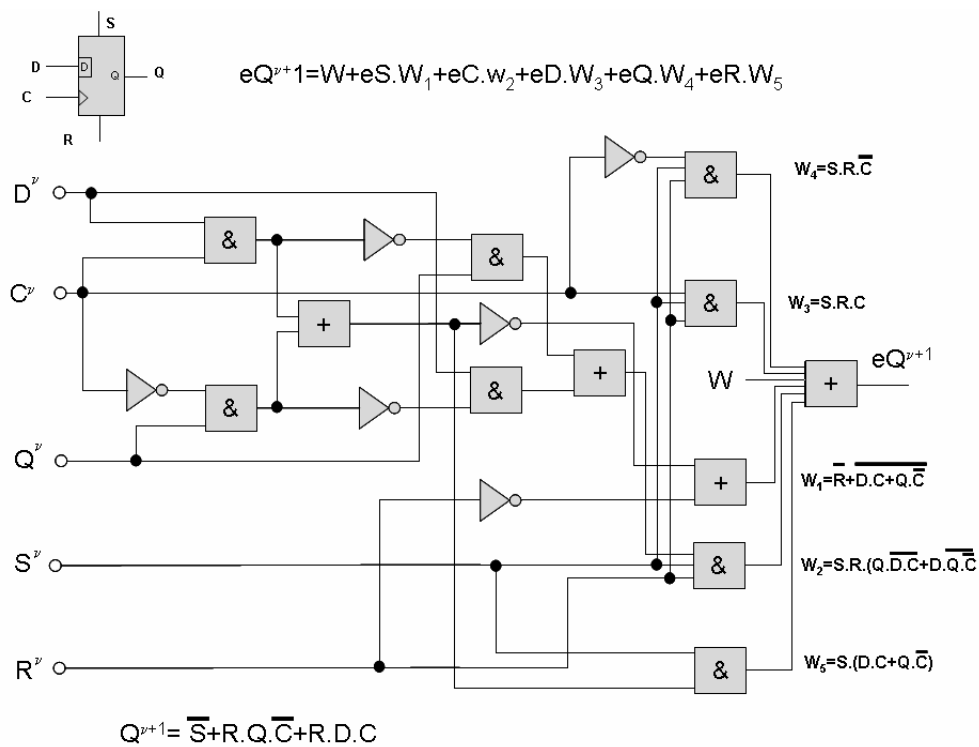


Fig.13 – Modelo de fallas FFD

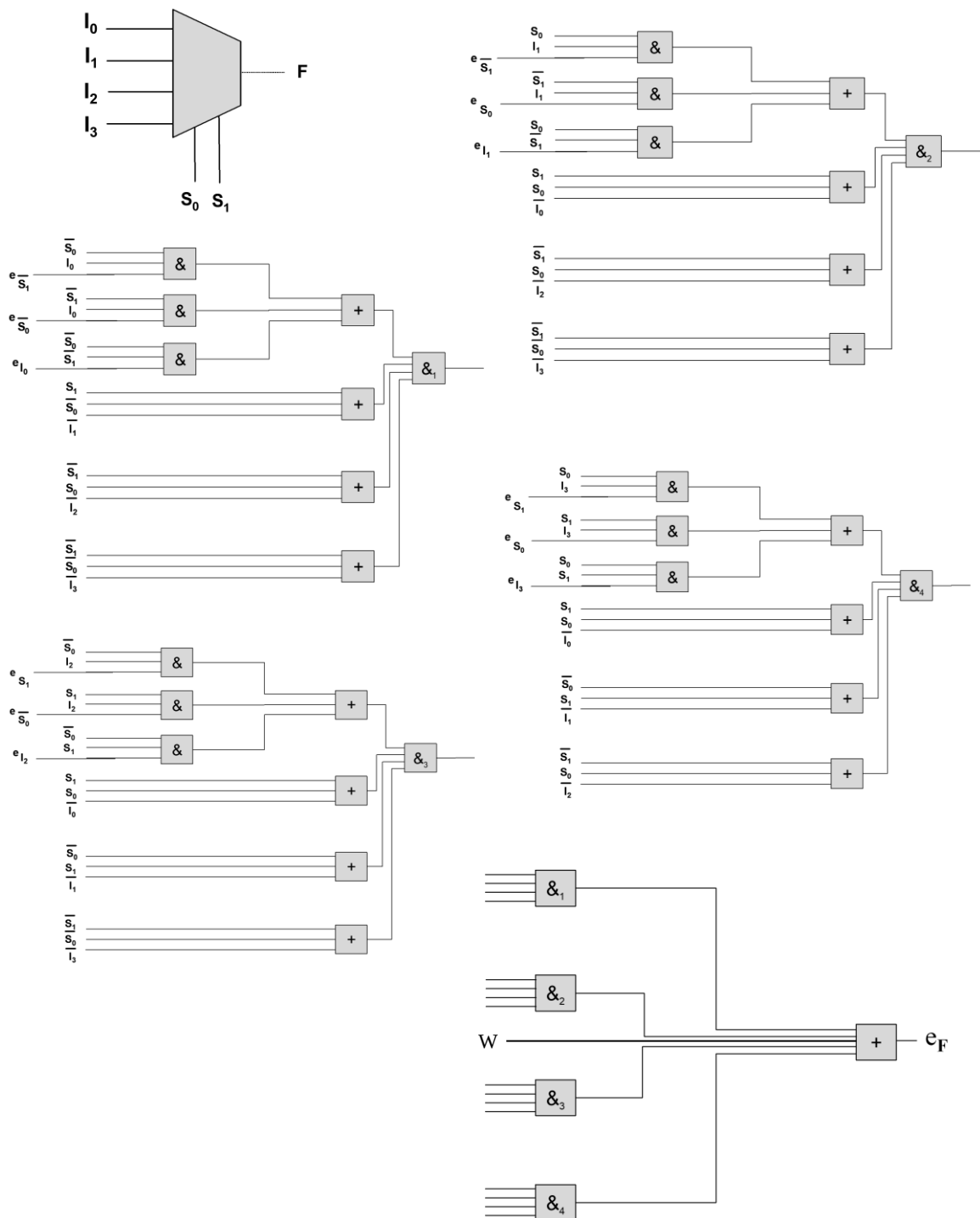


Fig.14 – Modelo de fallas MUX – 4 Entradas

6. Modelos de Falla de Bloques de FPGA genéricos

6.1. Modelo de falla de un CLB

El modelo de falla de un CLB_j depende de las fallas intrínsecas de cada componente a saber:

CLB_j Multiplexores = WMux_{31j}, WMux_{32j}, WMux_{33j}, WMux_{34j}, WMux_{35j}, WMux_{36j}, WMux_{37j}, WMux_{38j}, WMux_{21j}, WMux_{22j}, WMux_{23j}, WMux_{24j}, WMux_{11j}, WMux_{12j}, WMux_{01j}, WMux_{00j}.

CLB_j Biestable D = W_D

Del estado de las entradas y biestable del mismo a saber:

CLB_j Entradas = SR_{1j}, SR_{2j}, SR_{3j}, SR_{4j}, SR_{5j}, SR_{6j}, SR_{7j}, SR_{8j}, SR_{9j}, SR_{10j}, SR_{11j}, SR_{12j}, SR_{13j}, SR_{14j}, SR_{15j}, SR_{16j}, i_{0j}, i_{1j}, i_{2j}, i_{3j}, S_j, R_j, CK_j, Q_j.

y de los errores presentes en estas entradas y estados:

CLB_j Estados y Entradas = eSR_{1j}, eSR_{2j}, eSR_{3j}, eSR_{4j}, eSR_{5j}, eSR_{6j}, eSR_{7j}, eSR_{8j}, eSR_{9j}, eSR_{10j}, eSR_{11j}, eSR_{12j}, eSR_{13j}, eSR_{14j}, eSR_{15j}, eSR_{16j}, ei_{0j}, ei_{1j}, ei_{2j}, ei_{3j}, eS_j, eR_j, eCK_j, eQ_j.

En las Figs. 15.1 a 15.9 se puede observar el modelo de fallas vía FTA para un CLB_j genérico de un LUT de orden 4 (Fig.6).

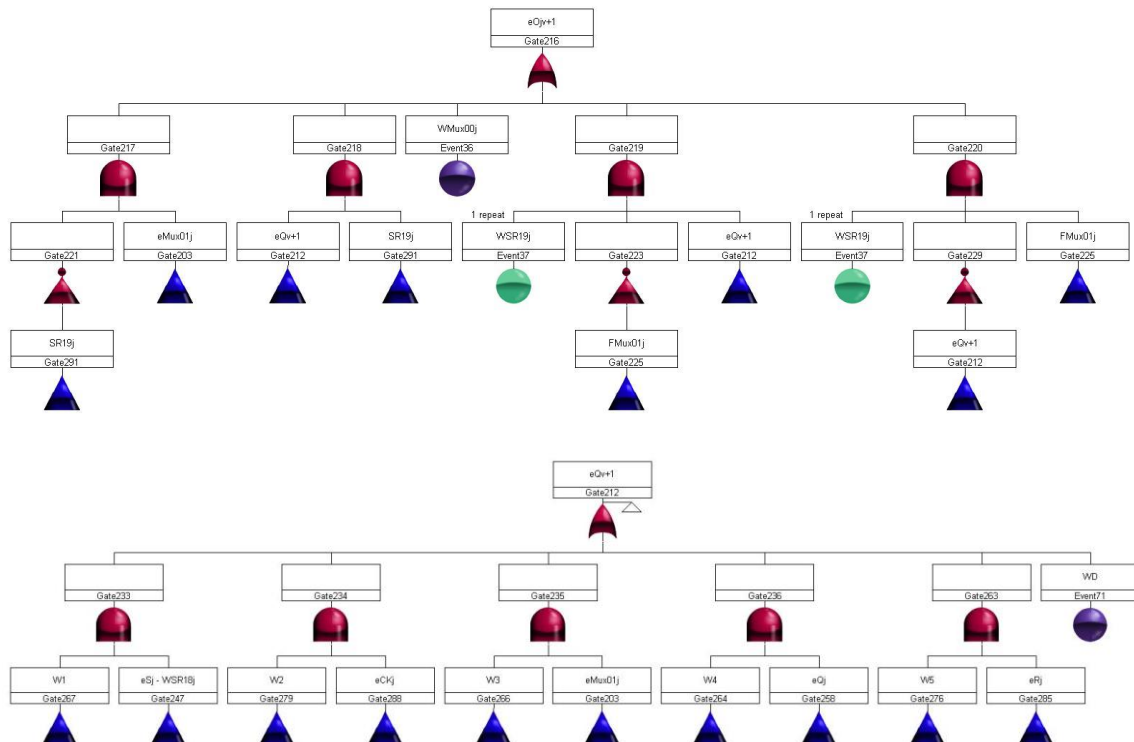


Fig.15.1 – Parcial del modelo de fallas de de un CLB genérico

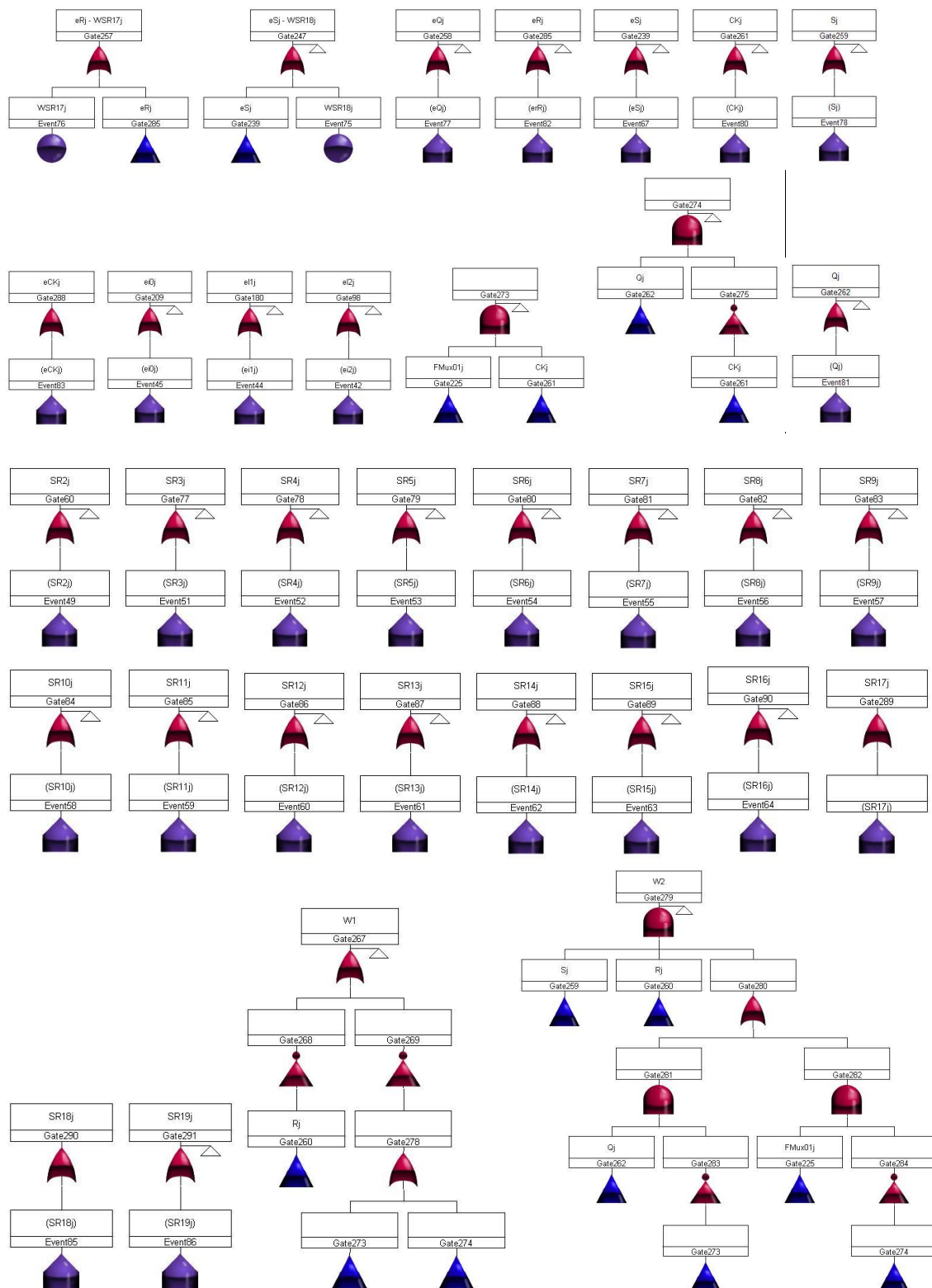


Fig.15.1 – Parcial del modelo de fallas de un CLB genérico

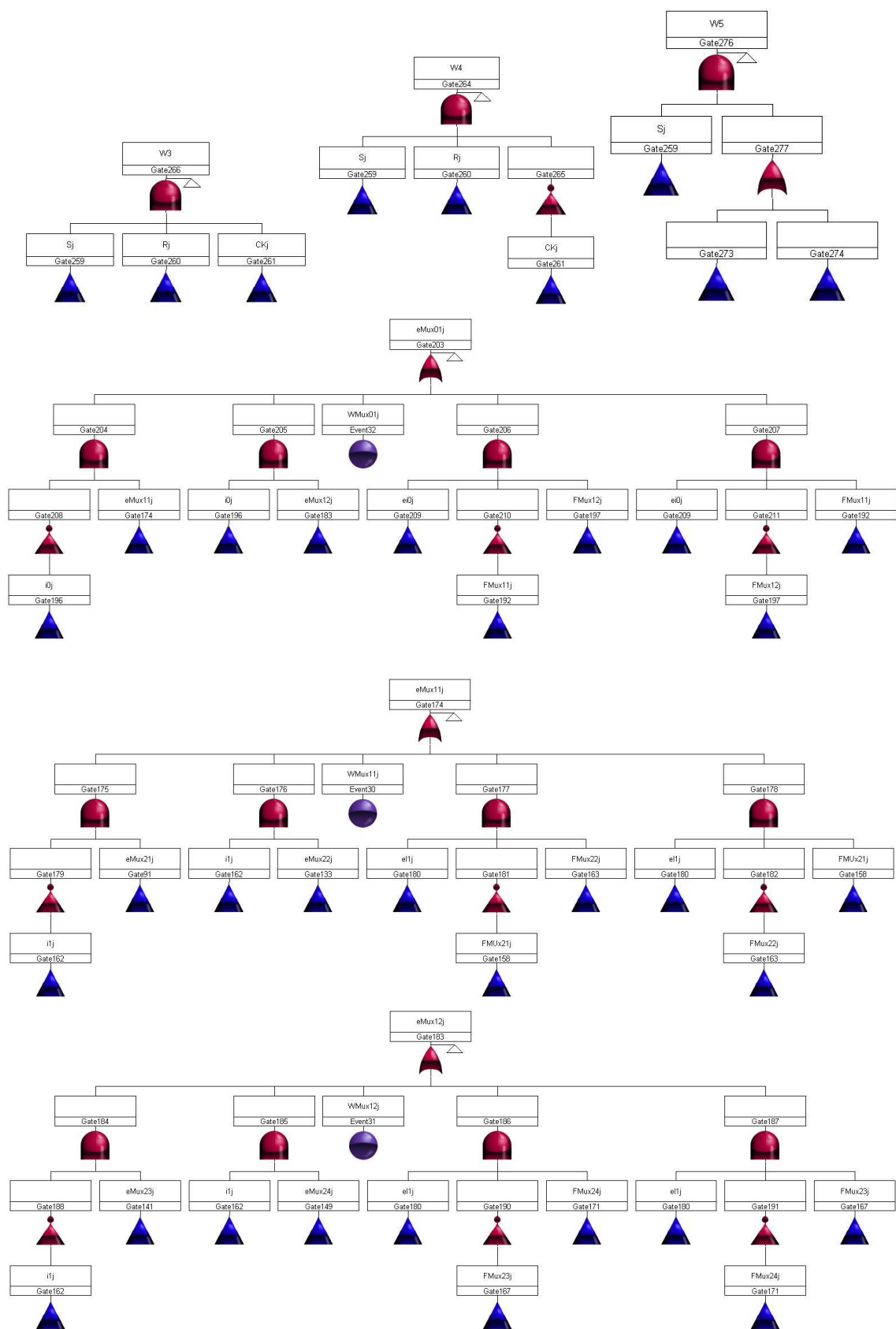


Fig.15.3 – Parcial del modelo de fallas de un CLB genérico

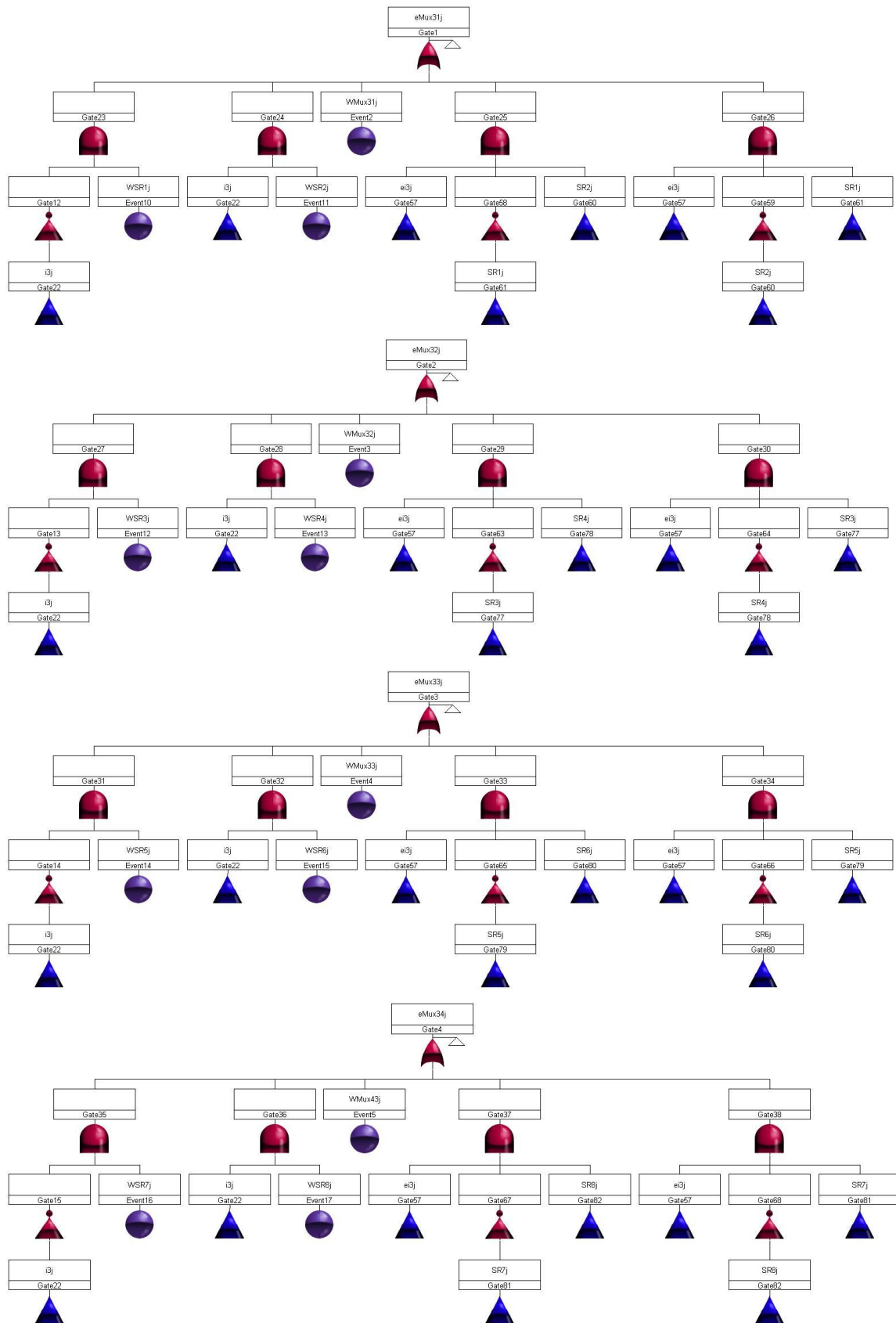


Fig.15.5 – Parcial del modelo de fallas de un CLB genérico

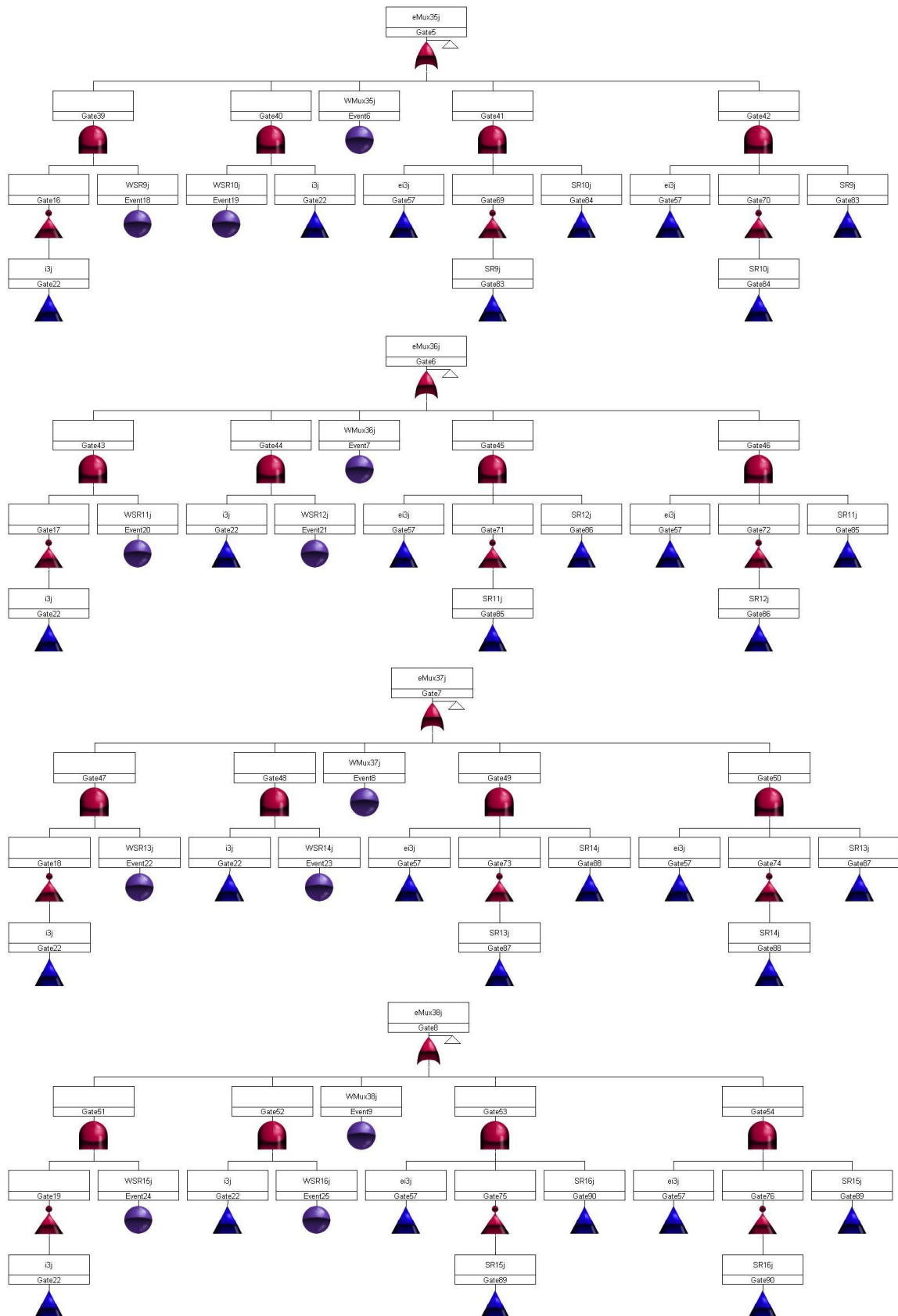


Fig.15.6 – Parcial del modelo de fallas de un CLB genérico

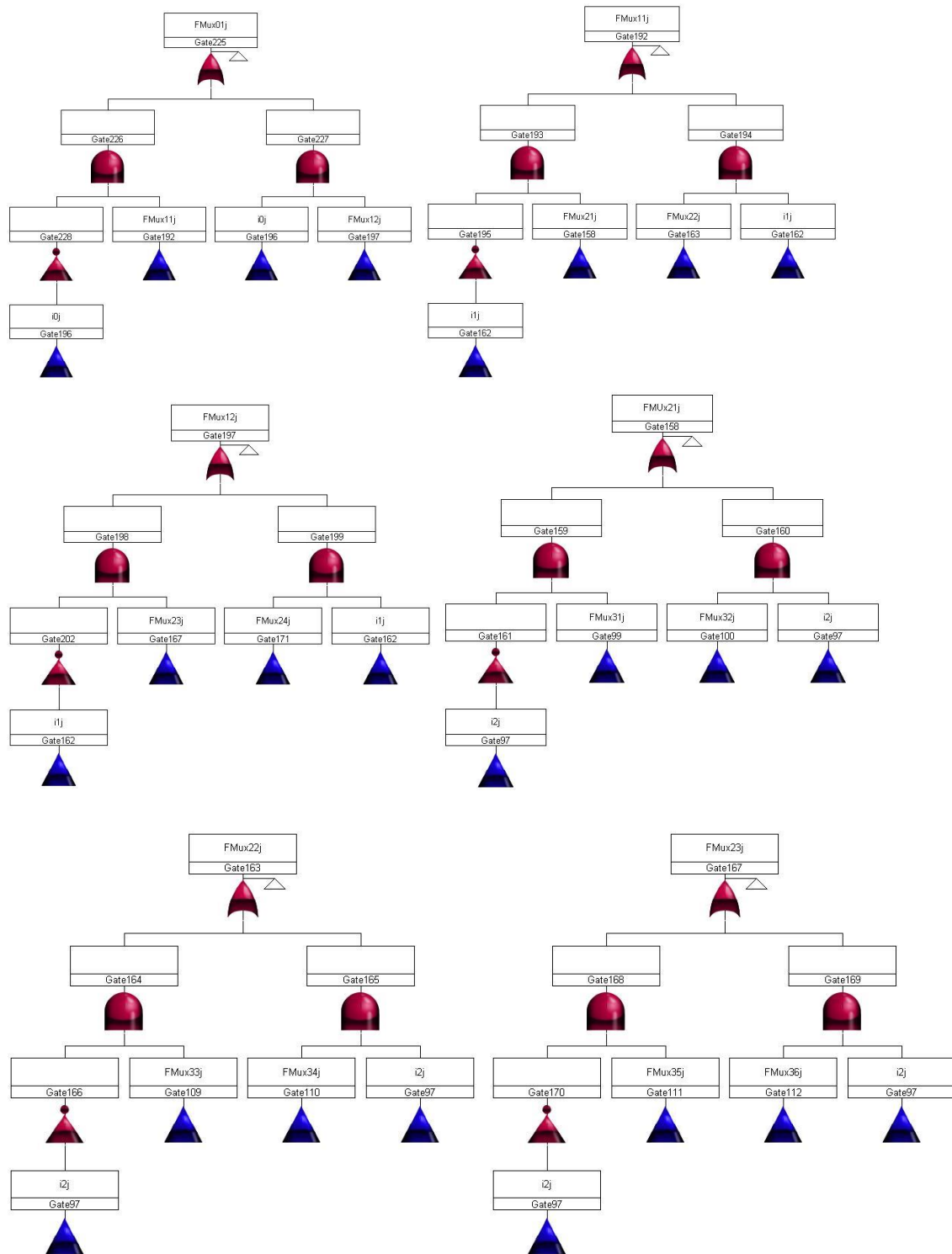
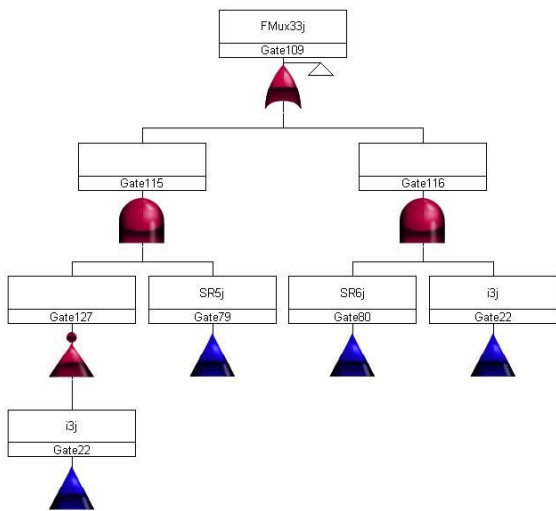
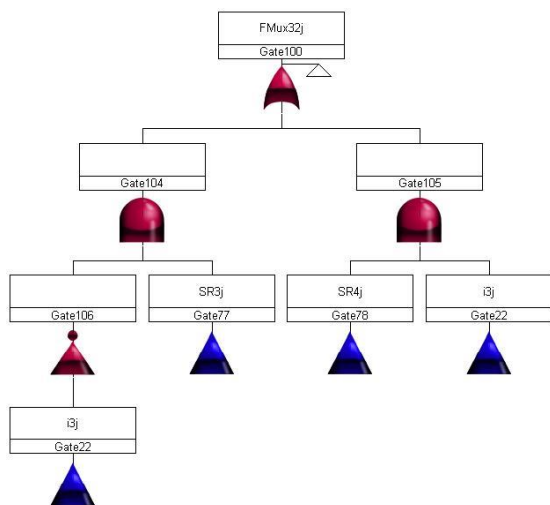
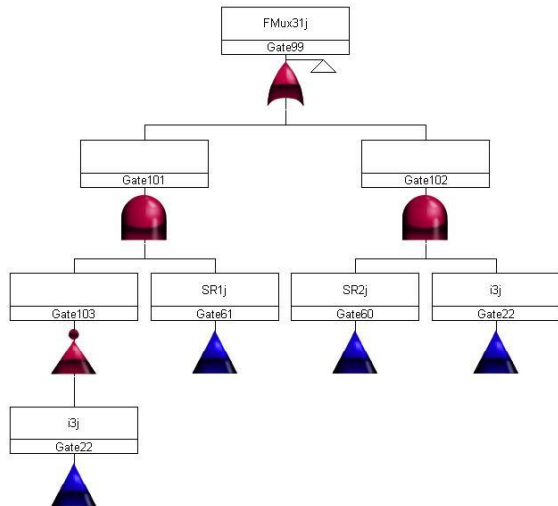
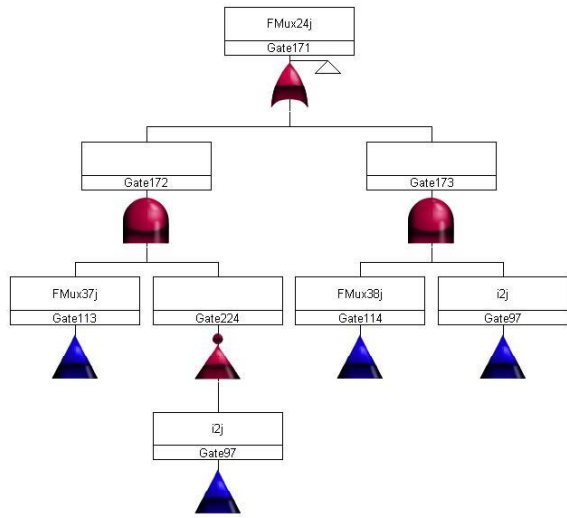


Fig.15.7 – Parcial del modelo de fallas de un CLB genérico



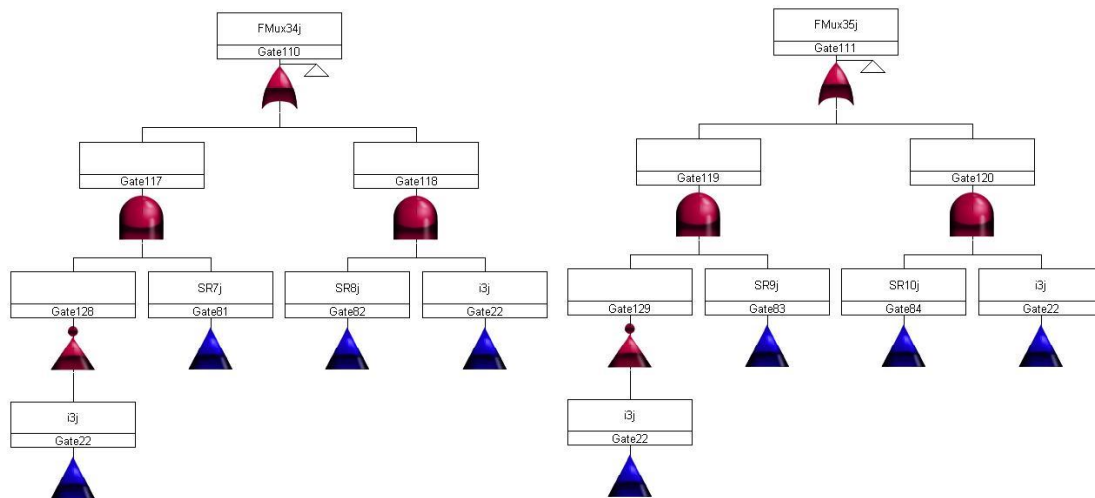


Fig.15.8 – Parcial del modelo de fallas de un CLB genérico

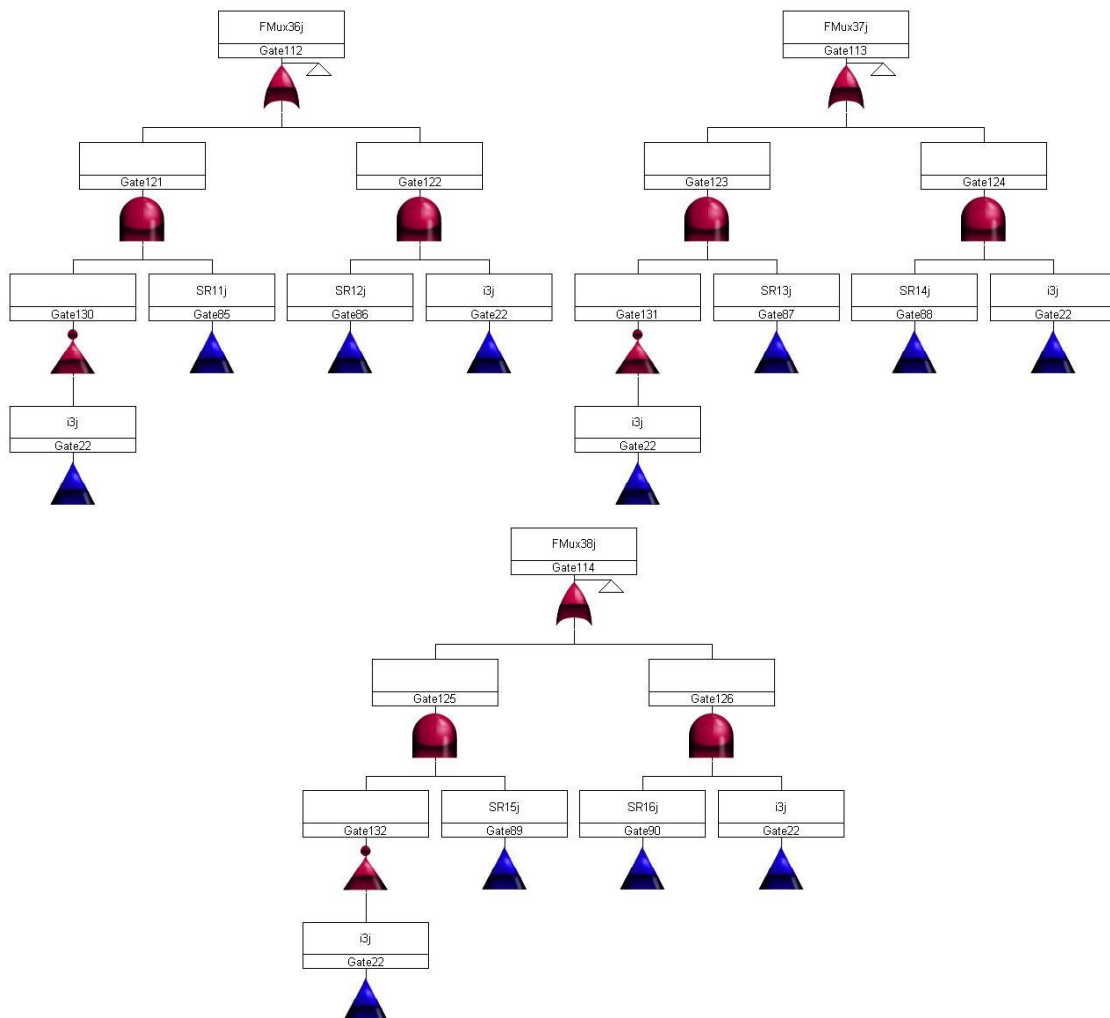


Fig.15.9 – Parcial del modelo de fallas de un CLB genérico

6.2. Modelo de falla de un SRAM

La constitución de una celda SRAM se muestra en las Figs.16 y 17

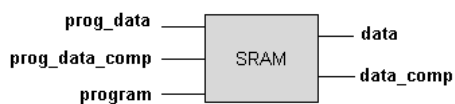


Fig.16 – Celda SRAM genérica

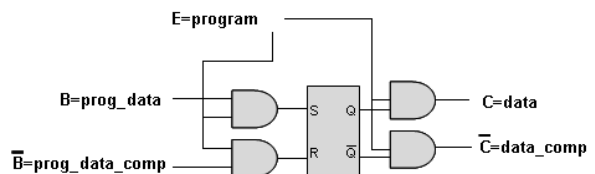


Fig.17 – Celda SRAM – Detalle

Básicamente consiste de dos compuertas inversoras acopladas y 2 compuertas de paso para programación. Para programar la celda se necesita del dato de programación y su complemento. En la Fig.18 se muestra el circuito equivalente al biestable RS utilizado cuya expresión es $Q^{v+1} = S + \bar{R} \cdot Q$

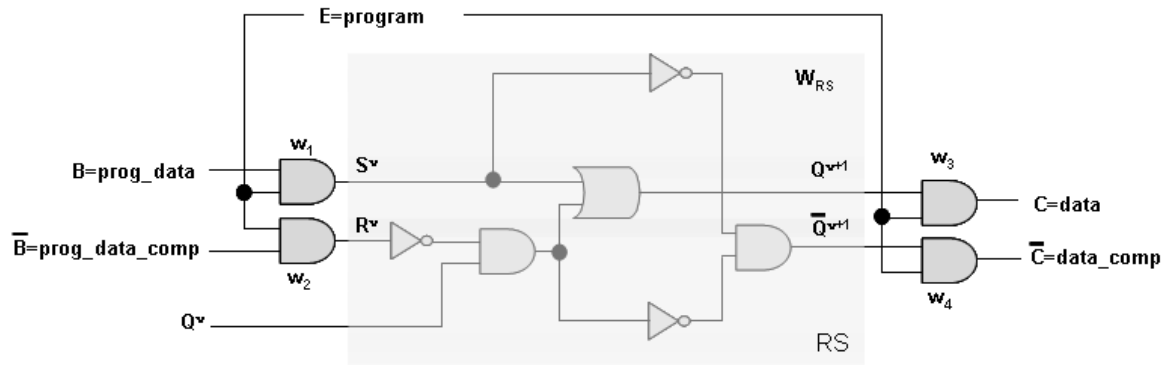


Fig.18 – Circuito equivalente biestable RS

El modelo de fallas básico para la SRAM se presenta en la Fig.19, donde W_{RS} , W_1 , W_2 , W_3 y W_4 son las probabilidades de falla del biestable RS y de las compuertas AND de entrada y salida.

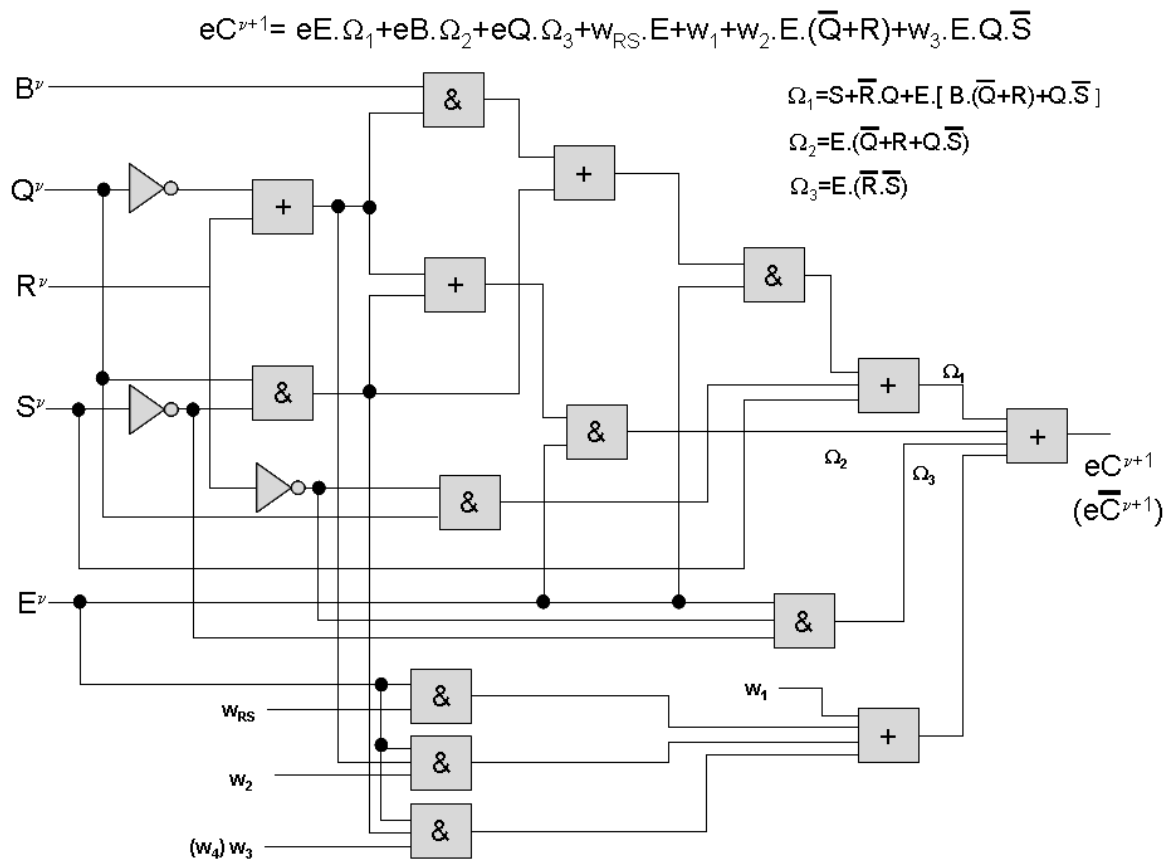


FIG.19 – Modelo de fallas básico de SRAM genérico

El modelo de árbol de fallas se muestra en las Figs. 20.1 a 20.2. El error en la salida complementaria $\bar{C}^{v+1}$ corresponde al mismo árbol de fallas donde se reemplaza $W3$ por $W4$. El modelo de falla de un SRAM_j depende de las fallas intrínsecas de cada componente a saber:

SRAM_j Compuertas = W_{1j} , W_{2j} , W_{3j} y W_{4j}

SRAM_j Biestable RS = W_{RSj}

Del estado de las entradas y del estado del biestable

SRAM_j Entradas = B_j , $\bar{B}_j$ y E_j

SRAM_j Biestable RS = Q_j

y de los errores presentes en estas entradas y estados:

SRAM_j Entradas y Estados = eB_j , eE_j , eQ_j

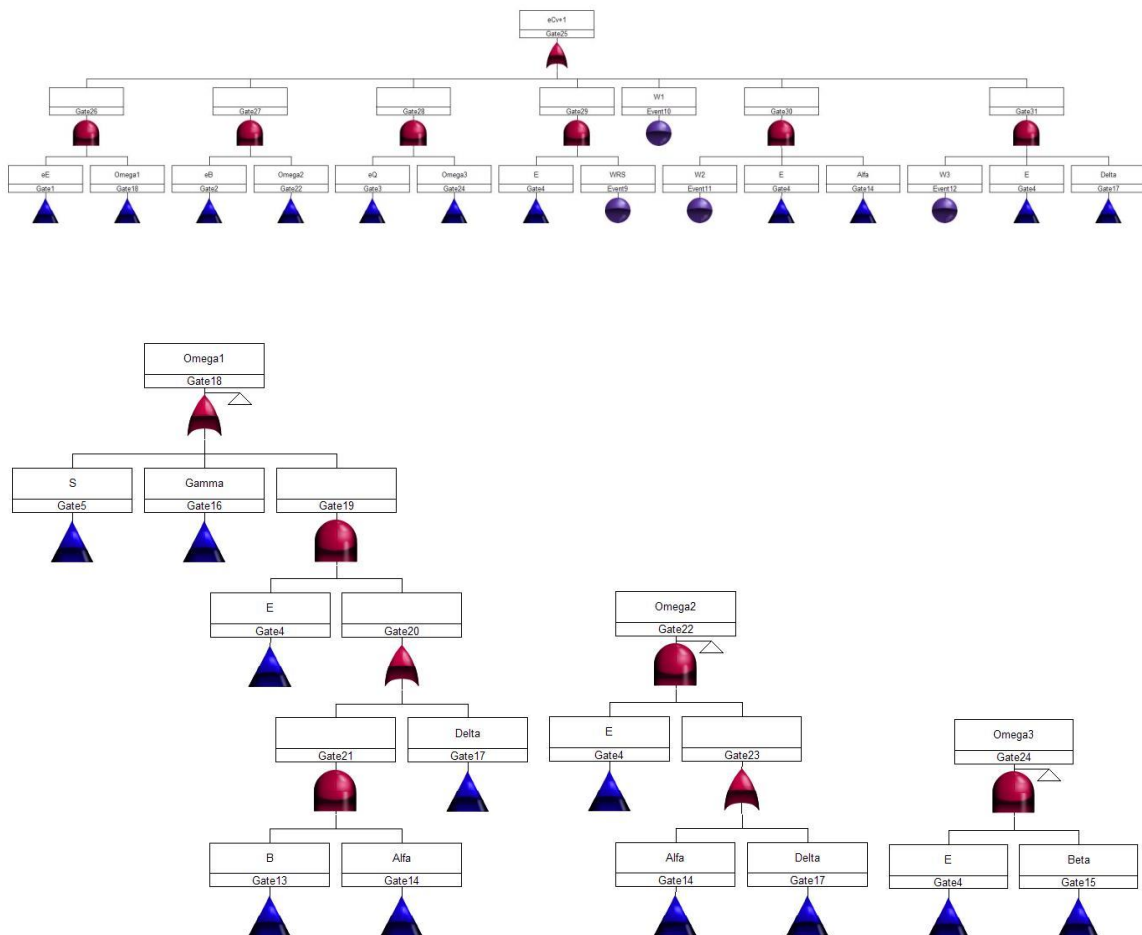


Fig.20.1 – Parcial del modelo de fallas de SRAM genérico

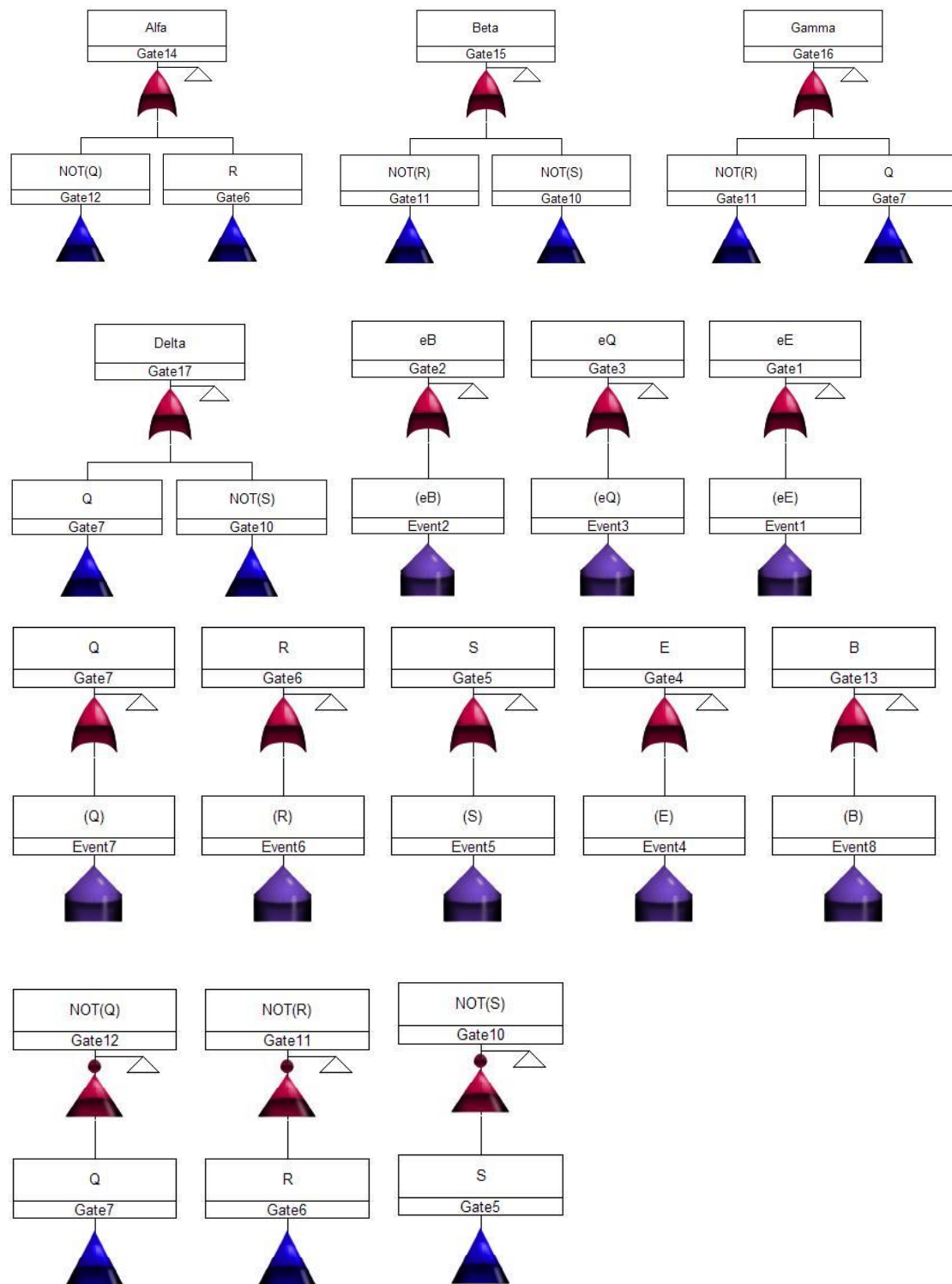


Fig.20.2 – Parcial del modelo de fallas de SRAM genérico

6.3. Modelo de falla de un Switch Block SW

En la Fig. 21 se observa una SW genérica utilizada como base de análisis con su tabla de verdad asociada. Esta SW esta conformada por dos llaves séxtuples comandadas por sendas SRAMs. La señal penetra por alguna de las cuatro entradas de una de las llaves séxtuples y puede seguir de largo o virar en una dirección u otra, de acuerdo al estado de cada uno de los transistores de paso. En la Fig. 21 se observa en el cuadro los estados de las llaves A, B, C, D, E y F suponiendo que la señal penetra en la SW desde la izquierda en la llave séxtuple superior. Asimismo en la Fig. 22 se muestran las distintas posibilidades de transito de la señal con la hipótesis de penetración de entrada por la izquierda en la llave séxtuple superior.

A	B	C	D	E	F
0	1	1	1	1	1
1	1	1	1	0	1
1	1	1	0	1	1
0	1	1	1	0	1
0	1	1	0	1	1
1	1	1	0	0	1
0	1	1	0	0	1

0 = Cerrado

1 = Abierto

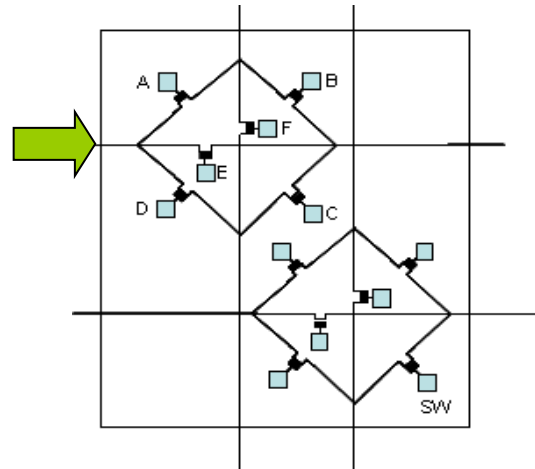


FIG.21 – Modelo de Switch Block genérico

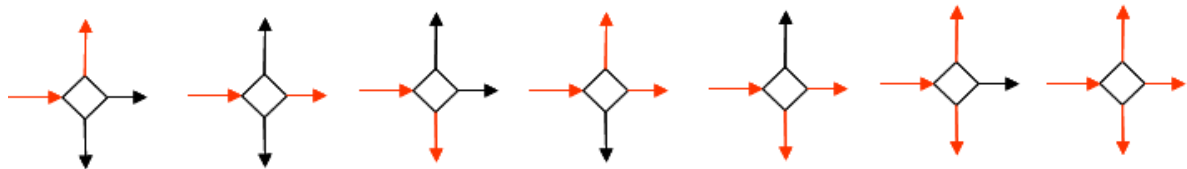


Fig.22 – Vías de ruteo y penetración de Switch Block genérico

Dos son las posibilidades de falla de los transistores de paso consideradas estadísticamente independientes.

w_a = Probabilidad de falla por abierto (Comando cerrar y opera abierto)

w_c = Probabilidad de falla por cerrado (Comando abrir y opera cerrado)

$1-w_a$ = Probabilidad de que no falle por abierto (Comando cerrar y opera cerrado)

$1-w_c$ = Probabilidad de que no falle por cerrado (Comando abrir y opera abierto)

$w_a + w_c - w_a \cdot w_c$ = Probabilidad de que falle

$1 - (w_a + w_c - w_a \cdot w_c)$ = Probabilidad de que no falle

Para los tres primeros casos de la Fig.22, la probabilidad de falla es la probabilidad de que falle por abierto el A (o falle su SRAM de comando) y al menos uno falle por cerrado de los B, C, D, E y F (o fallen sus SRAMs de comando). Eso es la probabilidad de que falle por abierto el A (o falle su SRAM

de comando) y el complemento de que ninguno de los B, C, D, E y F falle por cerrado (o fallen sus SRAMs de comando).

Para los tres segundos casos de la Fig. 22, la probabilidad de falla es la probabilidad de que falle por abierto el A y el E (o fallen sus SRAMs de comando) y al menos uno falle por cerrado de los B, C, D y F (o fallen sus SRAMs de comando). Esto es la probabilidad de que falle por abierto el A y el E (o fallen sus SRAMs de comando) y el complemento de que ninguno de los B, C, D y F falle por cerrado (o fallen sus SRAMs de comando)

Finalmente para el ultimo caso de la Fig.22, la probabilidad de falla es la probabilidad de que falle por abierto el A, E y el D (o fallen sus SRAMs de comando) y al menos uno falle por cerrado de los B, C y F (o fallen sus SRAMs de comando). Esto es la probabilidad de que falle por abierto el A, E y el D (o fallen sus SRAMs de comando) y el complemento de que ninguno de los B, C y F falle por cerrado (o fallen sus SRAMs de comando).

En las Figs.23.1 a 23.3 se muestra el modelo de fallas para la llave séxtuple superior de la Fig. 21 con la entrada de señal desde la izquierda y con Top Gate SWLIF.

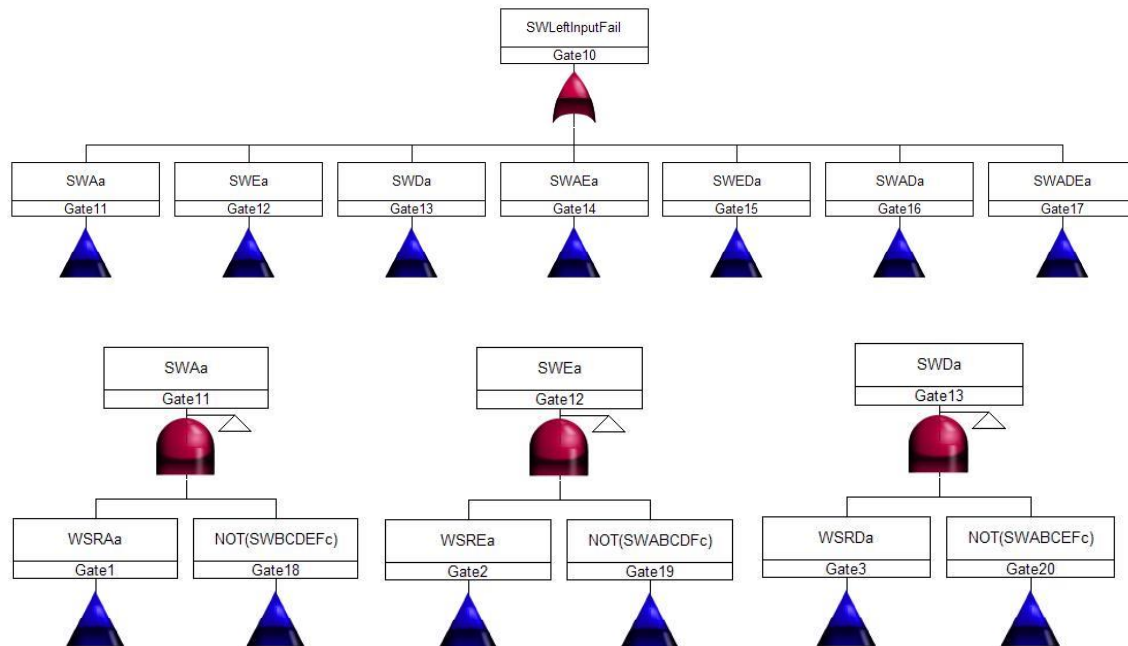


Fig.23.1 – Parcial del modelo de fallas de Switch Block genérico

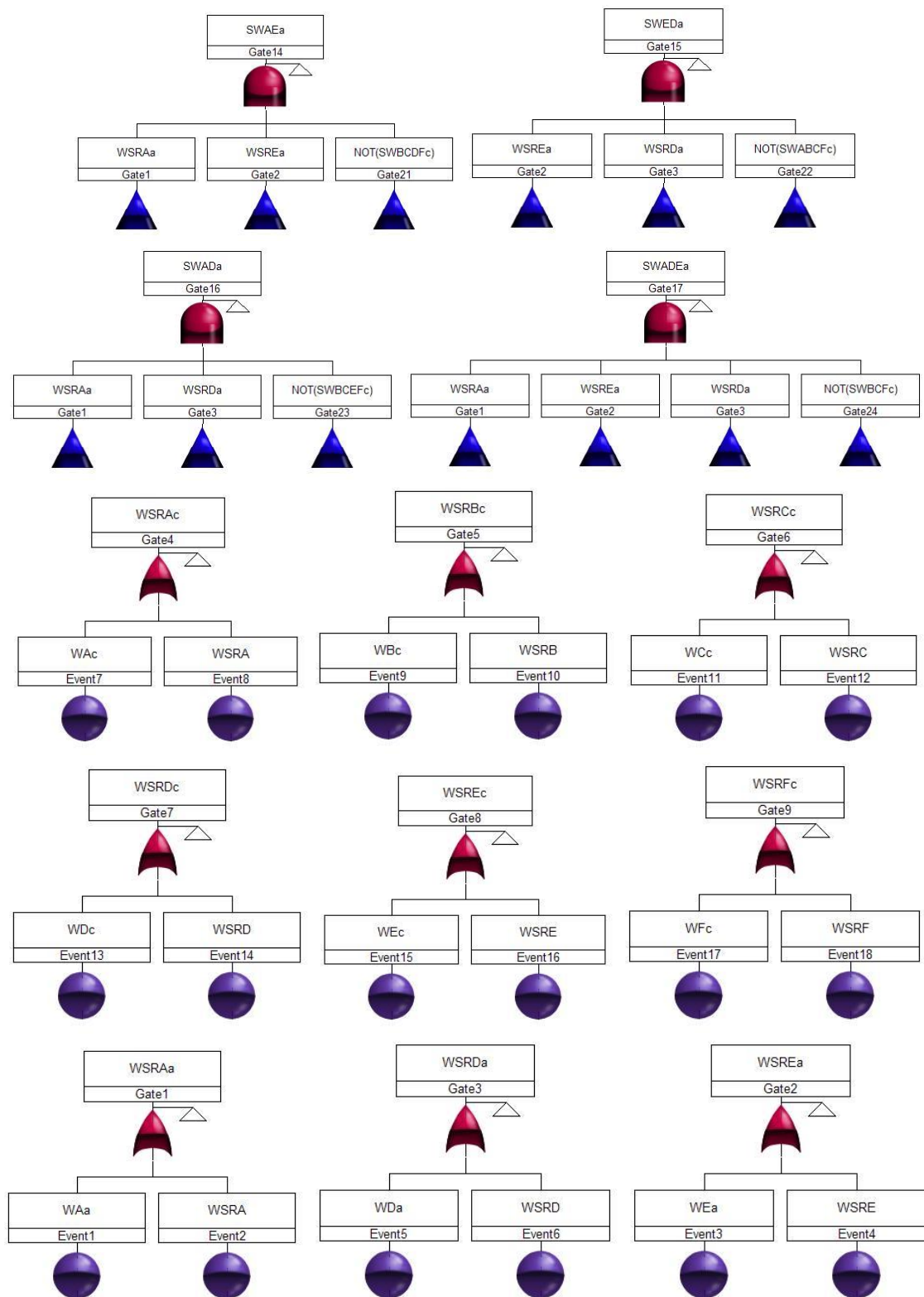


Fig.23.2 – Parcial del modelo de fallas de Switch Block genérico

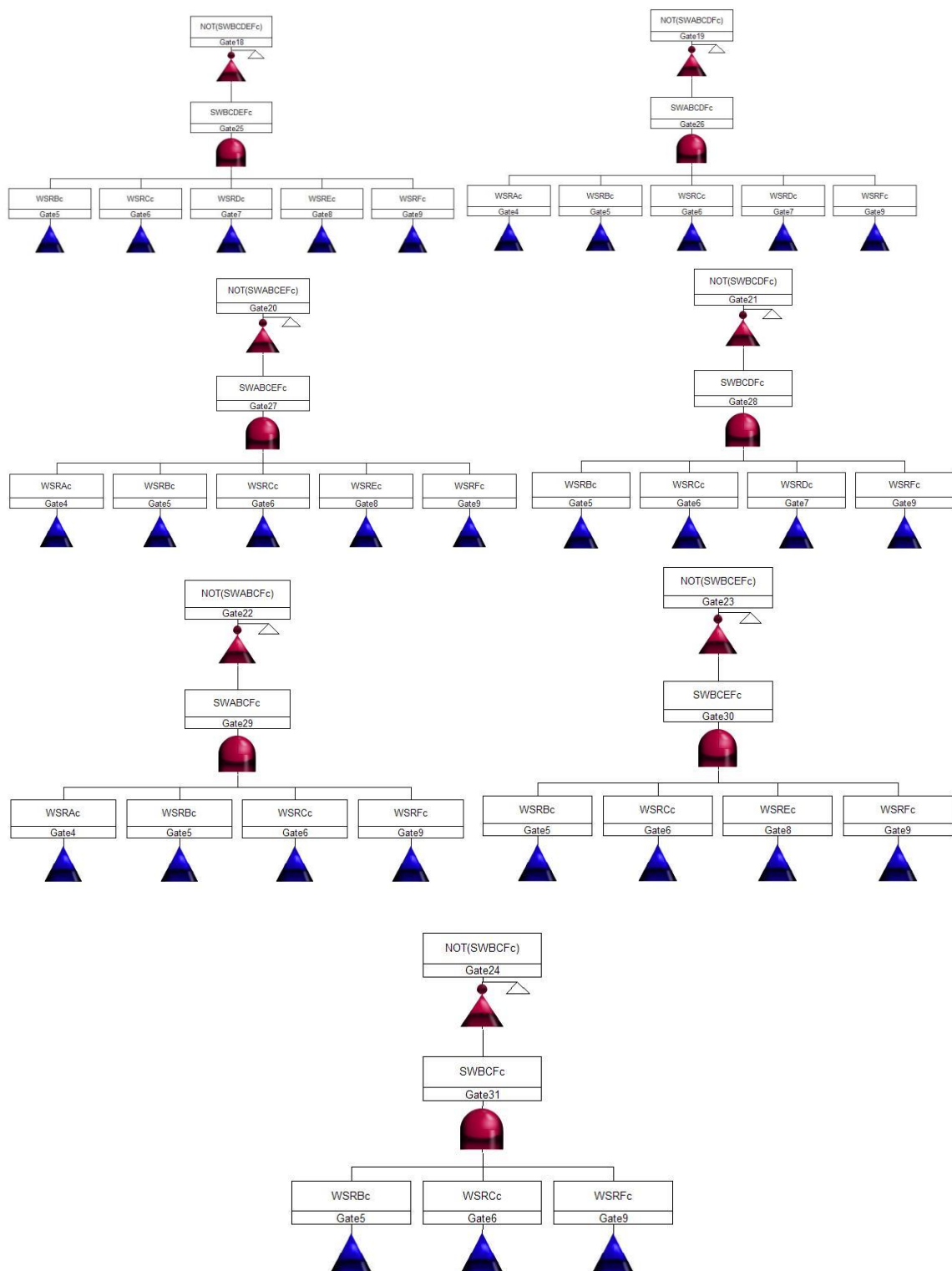


Fig.23.3 – Parcial del modelo de fallas de Switch Block genérico

El modelo de falla de la llave séxtuple superior de la Fig. 21 con la entrada de señal desde la izquierda SWLIF dependerá de las fallas intrínsecas de cada componente de esa llave, a saber:

WAc, WBc, WCc, WDC, WEc, WFc = falla por Cerrado de los transistores de paso asociados a las ramas A, B, C, D, E y F de la llave séxtuple

WSRA, WSRB, WSRC, WSRD, WSRE, WSRF = falla de la SRAM asociada al comando del transistor de paso de las ramas A, B, C, D, E y F de la llave séxtuple

WAa, WDa, WEa = falla por Abierto de los transistores de paso asociados a las ramas A, D y E de la llave séxtuple

Los modelos para las distintas combinaciones, esto es con señal entrando desde arriba, desde abajo o desde la derecha son similares. Así, en forma similar al árbol de fallas con Top Gate SWLIF se tendrán árboles de fallas con Top Gates SWAIF, SWBIF y SWRIF para cada una de las llaves séxtuples ya sea la superior o la inferior. En total se tendrán ocho árboles de falla, cuatro para la llave séxtuple superior y cuatro para la inferior de la Fig. 21.

6.4. Modelo de falla de un Bloque Input-Output

En la Fig.24 se observa un bloque de entrada salida genérico.

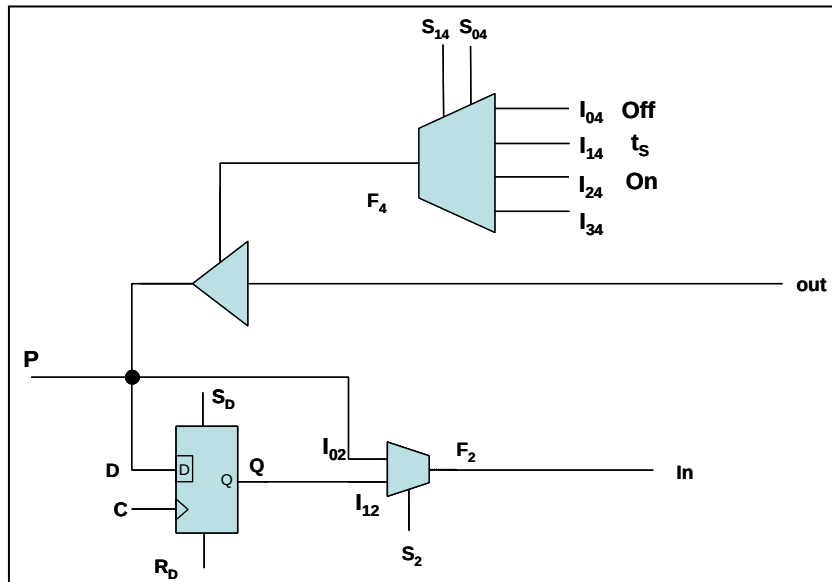


Fig.24 – Modelo del Bloque Input – output genérico

Con este modelo y los expuestos precedentemente para el bloque de entrada salida el modelo completo de falla del bloque resulta con las expresiones siguientes:

$$in = (\overline{S_D} + R_D \cdot Q^v \cdot \overline{C} + R_D \cdot D^v \cdot C) \cdot S_2 + \overline{S_2} \cdot P$$

$$P = D^v = out \cdot (Off \cdot \overline{S_{14}} \cdot \overline{S_{04}} + t_s \cdot \overline{S_{14}} \cdot S_{04} + On \cdot S_{14} \cdot \overline{S_{04}})$$

El modelo de falla de un bloque Input-Output depende de las fallas intrínsecas de cada componente a saber:

Multiplexores = WM₄, WM₂

Biestable D = W_D

Del estado de las entradas y biestable del mismo a saber:

I/O Entradas = SR₁₄, SR₀₄, SR₂, WSR₁₄, WSR₂, WSRR, WSRS, S, R, CK, D, Q.

y de los errores presentes en estas entradas y estados:

I/O Estados y Entradas = eS₁₄, eS₀₄, eS₂, eS, eR, eCK, eQ, eD, eI_{OFF}, eI_{ts}, eI_{ON}, eI_{FREE}, e_{OUT}

En las Fig. 25.1 a 25.6 se muestran los árboles de falla correspondientes.

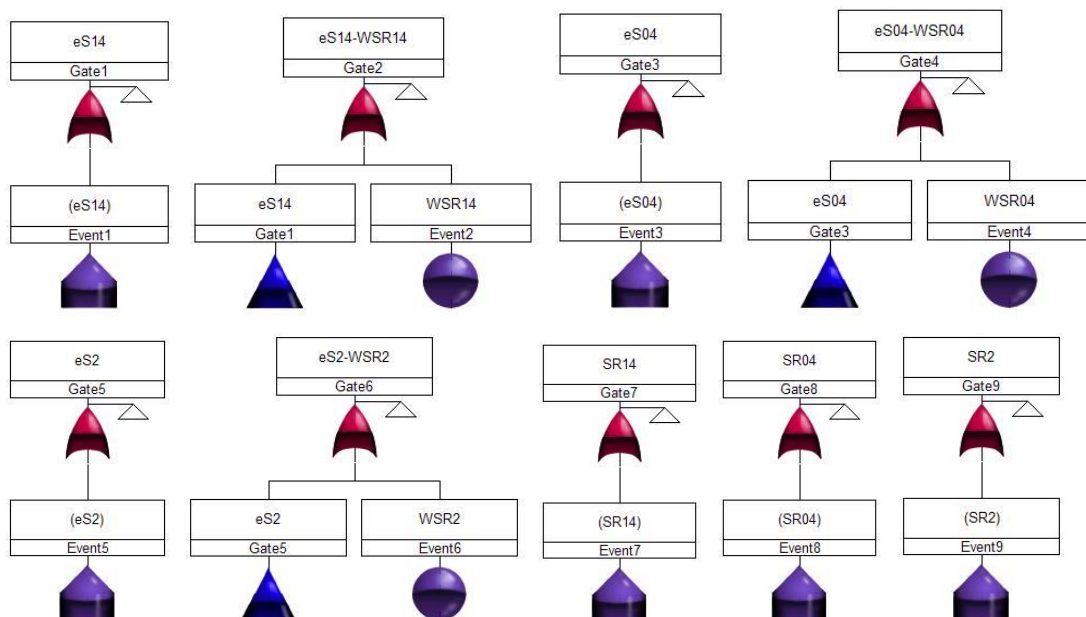


Fig.25.1 – Parcial del modelo de fallas del bloque Input – Output genérico

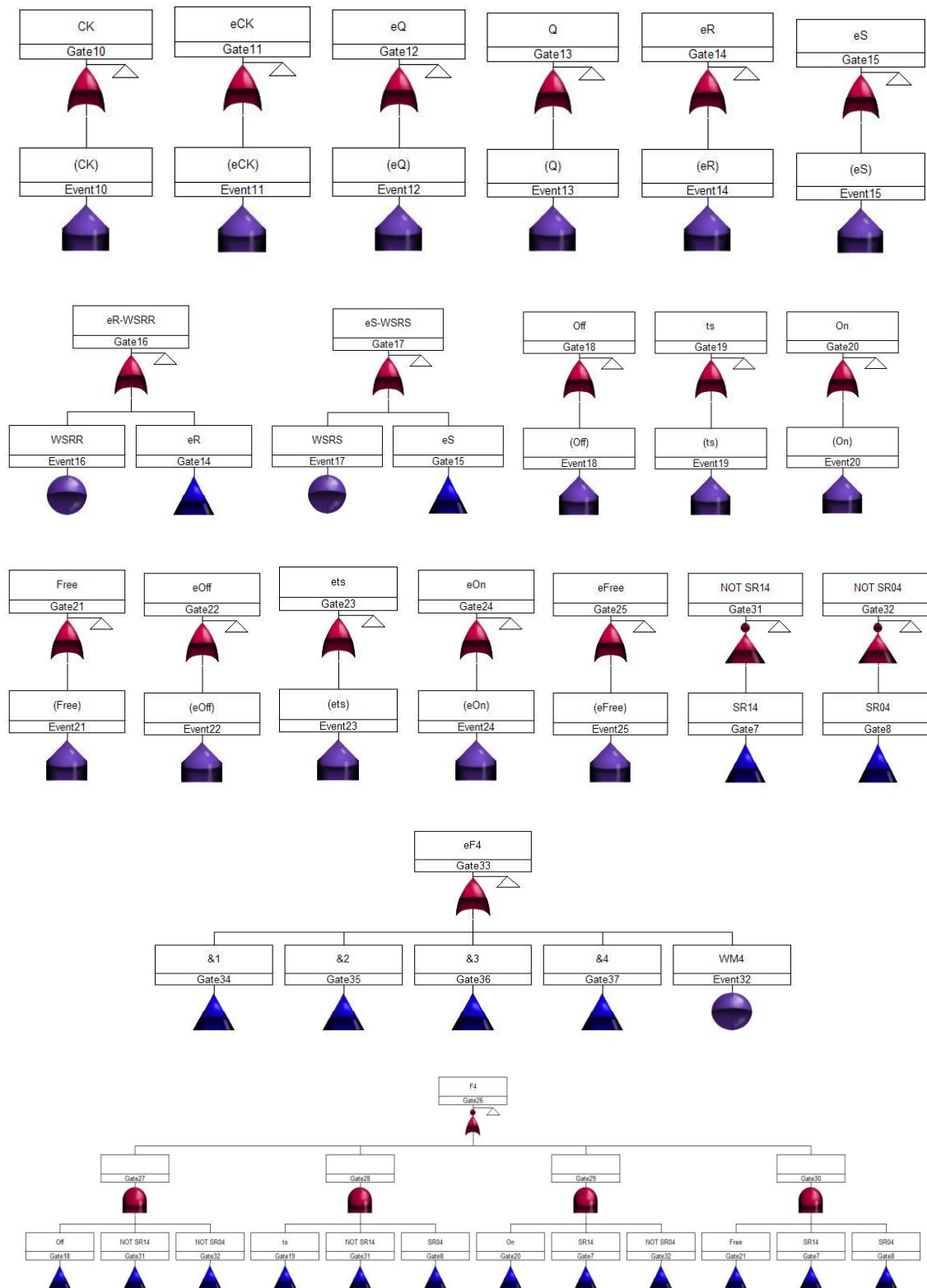


Fig.25.2 – Parcial del modelo de fallas del bloque Input – Output genérico

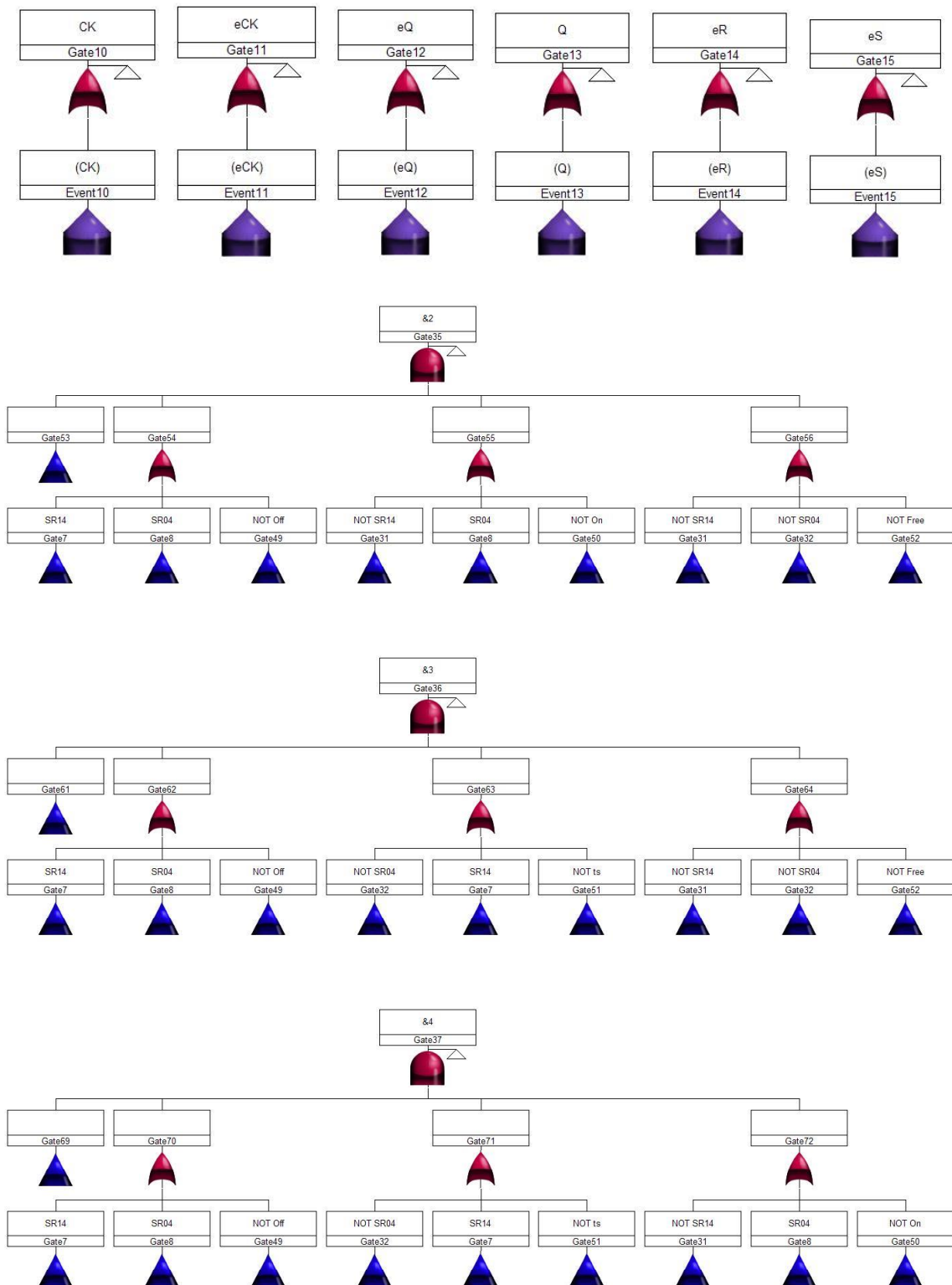


Fig.25.3 – Parcial del modelo de fallas del bloque Input – Output genérico

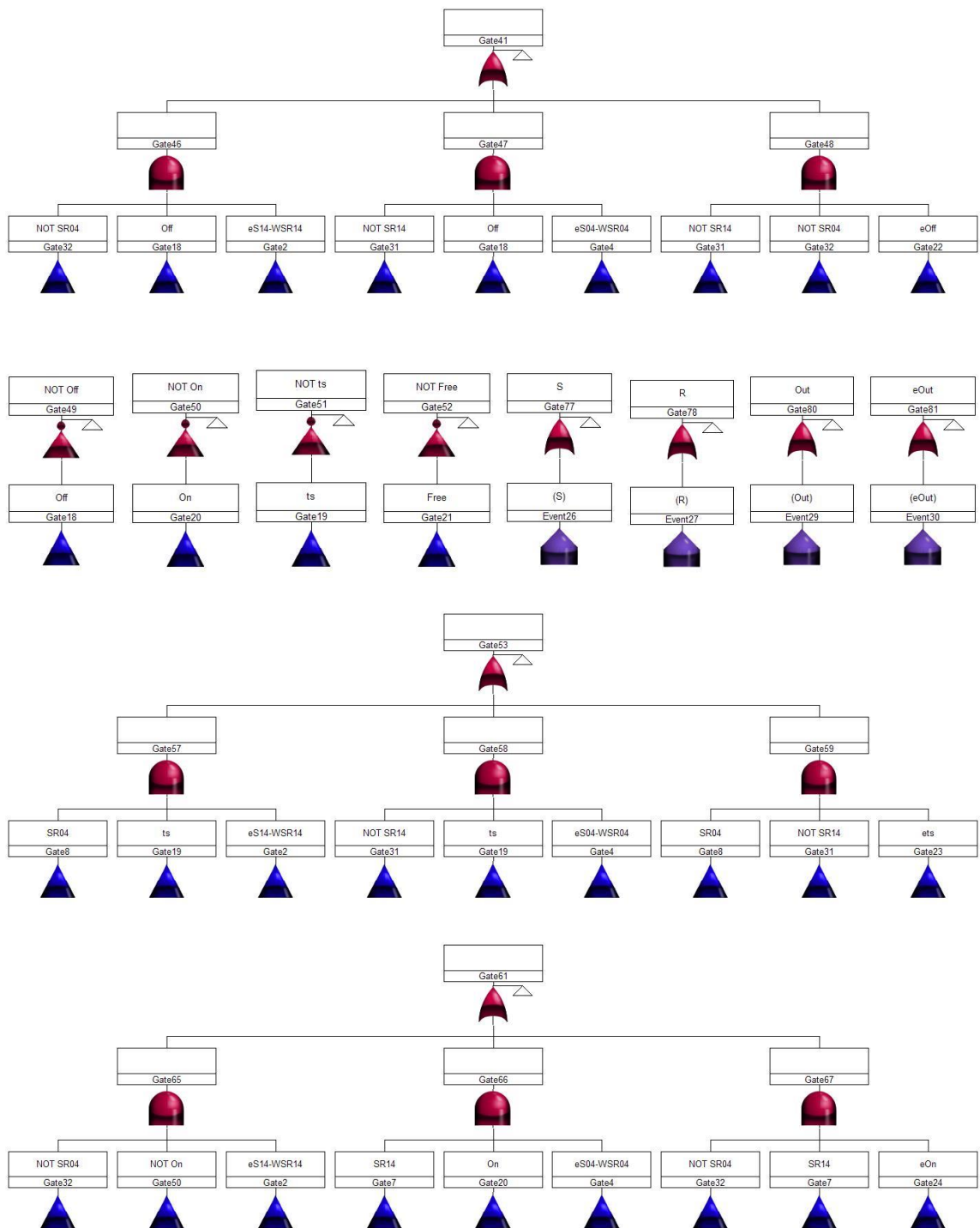


Fig.25.4 – Parcial del modelo de fallas del bloque Input – Output genérico

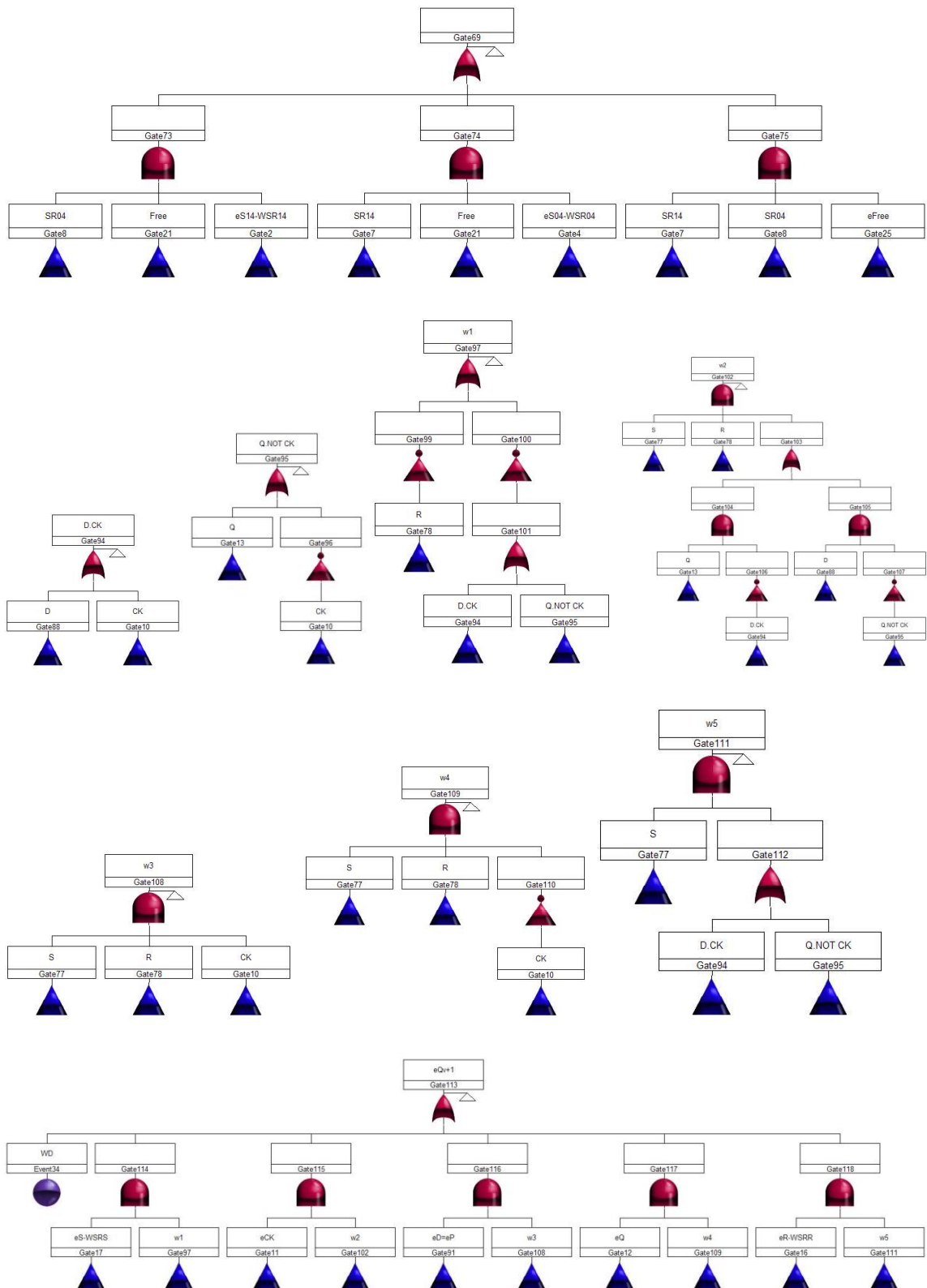


Fig.25.5 – Parcial del modelo de fallas del bloque Input – Output genérico

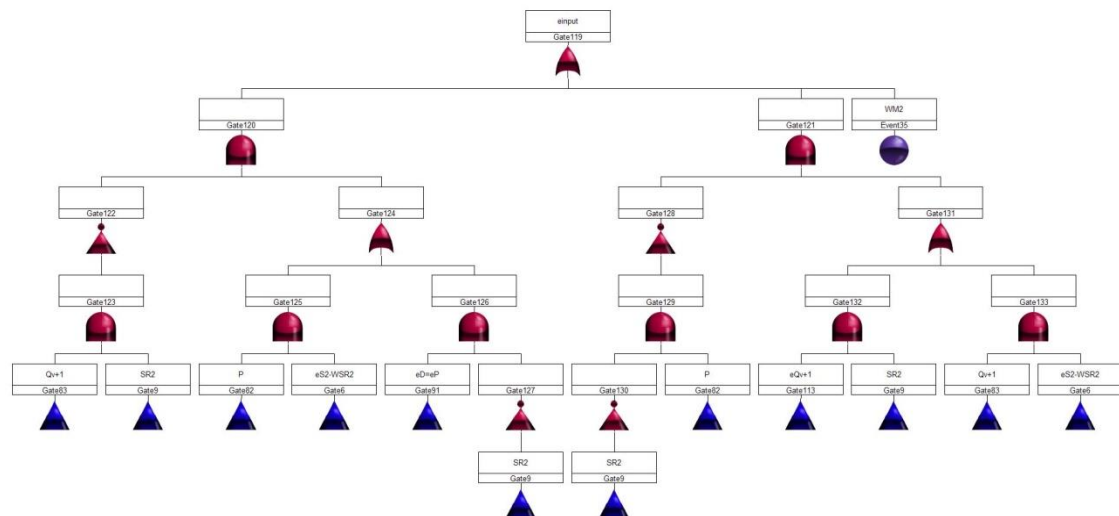


Fig.25.6 – Parcial del modelo de fallas del bloque Input – Output genérico

6.5. Modelo de falla de un Bloque de conexión CB_1

En la Fig.26 se muestra el bloque genérico correspondiente al bloque colocado a la derecha (right) de un CLB con las líneas verticales de conexión interna.

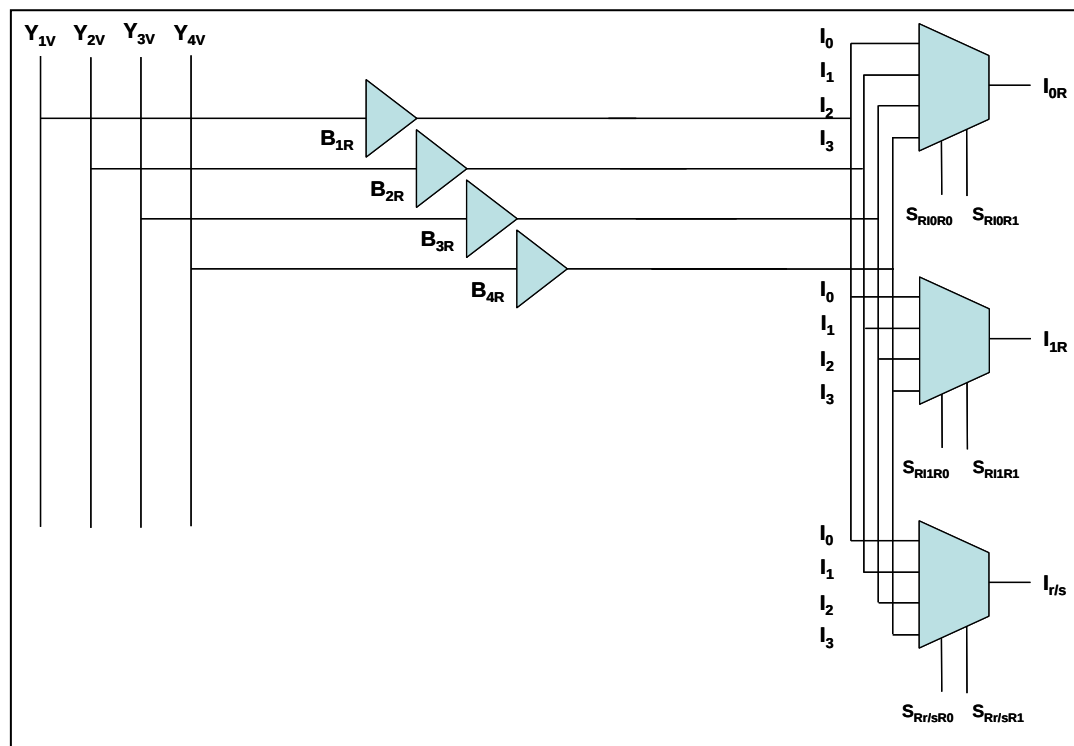


Fig.26 – Modelo del bloque de interconexión CB_1 genérico

El modelo de falla de un bloque CB_1 depende de las fallas intrínsecas de cada componente a saber:

Multiplexores = $WI0R$, $WI1R$, Wr/sR

Buffers = $BR3$, $BR4$, $B2R$, $B1R$

Del estado de las entradas y biestable del mismo a saber:

CB_1 Entradas = $SRI0R0$, $SRI0R1$, $SRI1R0$, $SRI1R1$, $Y1V$, $Y2V$, $Y3V$, $Y4V$

y de los errores presentes en estas entradas y estados:

I/O Estados y Entradas = $eSRI0R0$, $eSRI0R1$, $eSRI1R0$, $eSRI1R1$, $eY1V$, $eY2V$, $eY3V$, $eY4V$

En las figuras 27.1 a 27.5 se muestran los árboles de falla correspondientes.

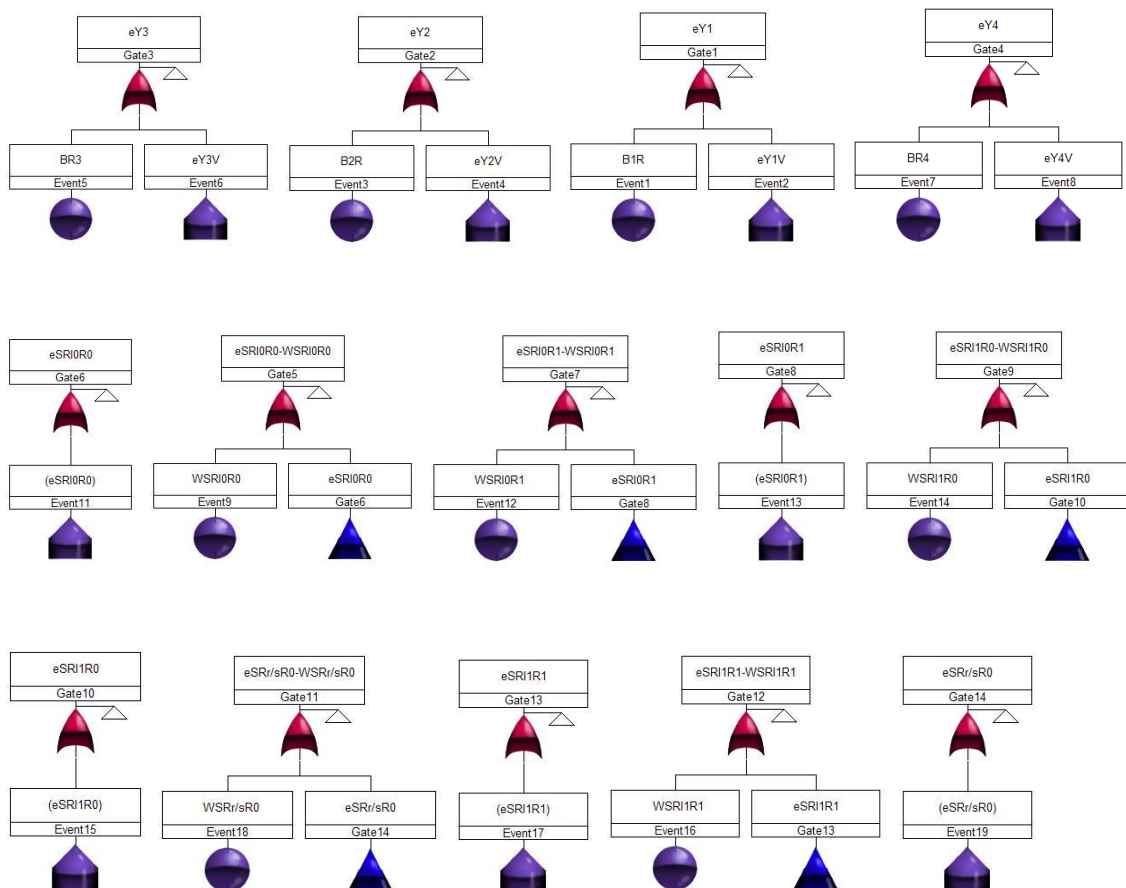


Fig.27.1 – Modelo de fallas del bloque de interconexión CB_1 genérico

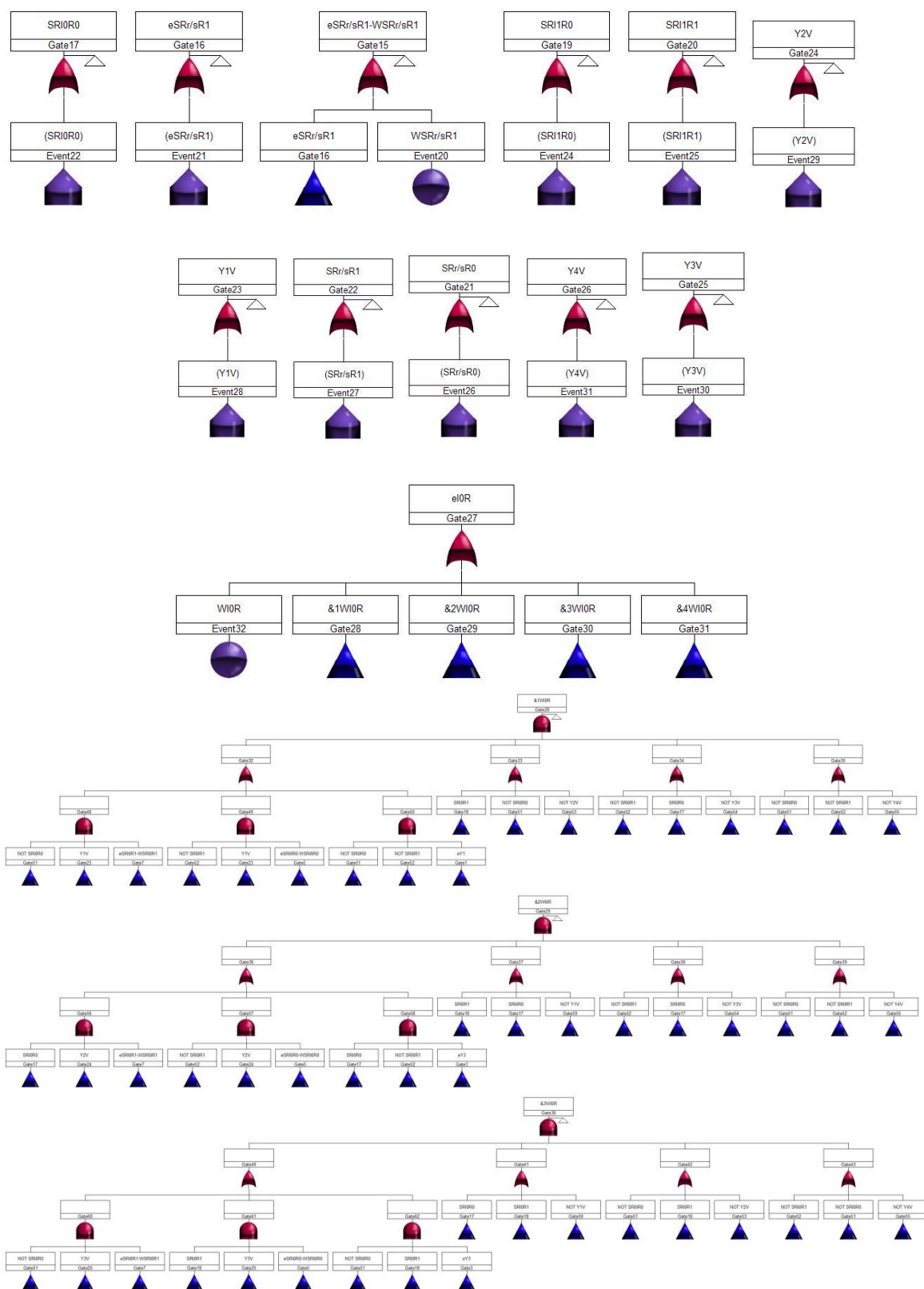


Fig.27.2 – Modelo de fallas del bloque de interconexión CB₁ genérico

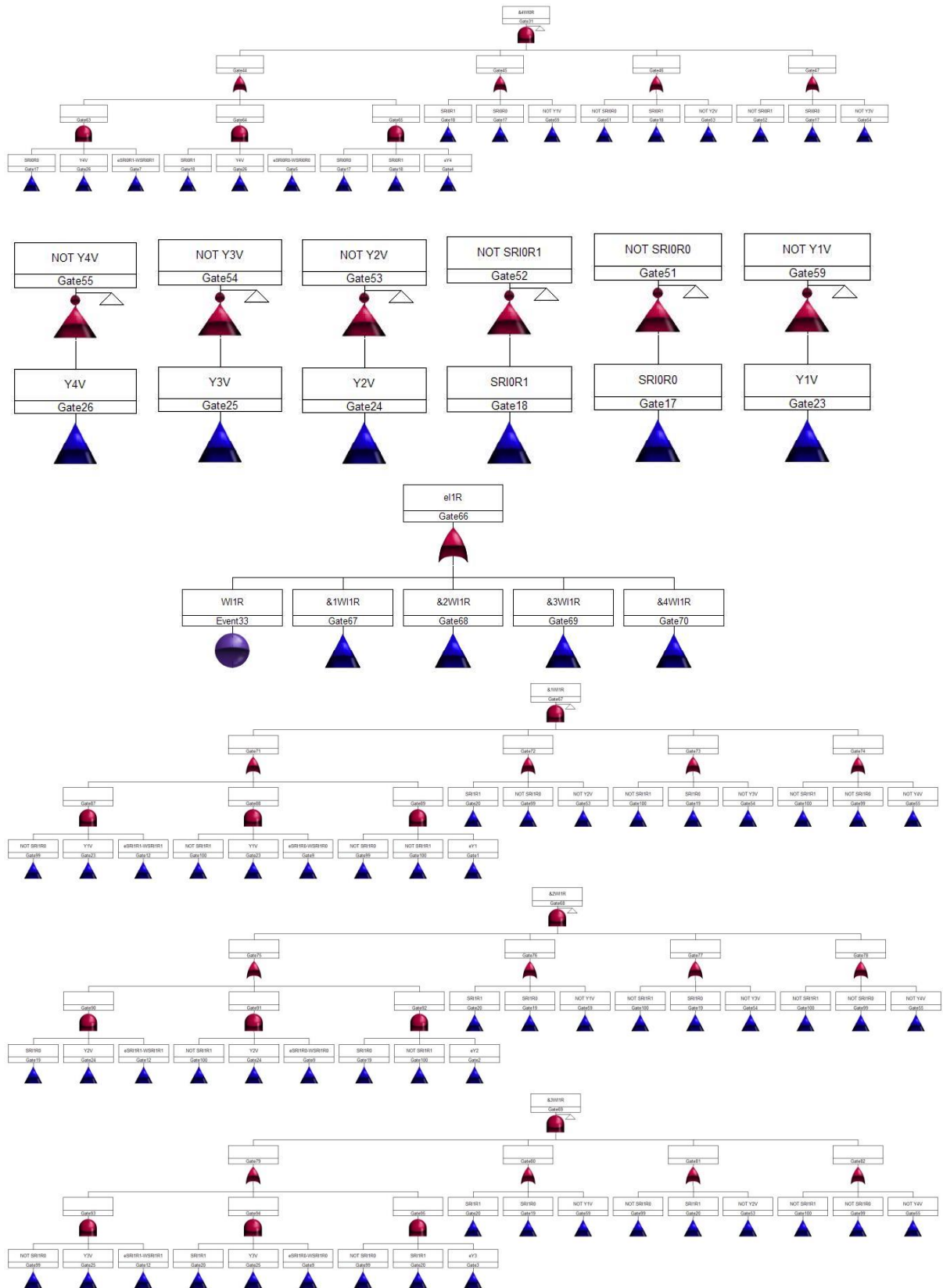


Fig.27.3 – Modelo de fallas del bloque de interconexión CB₁ genérico

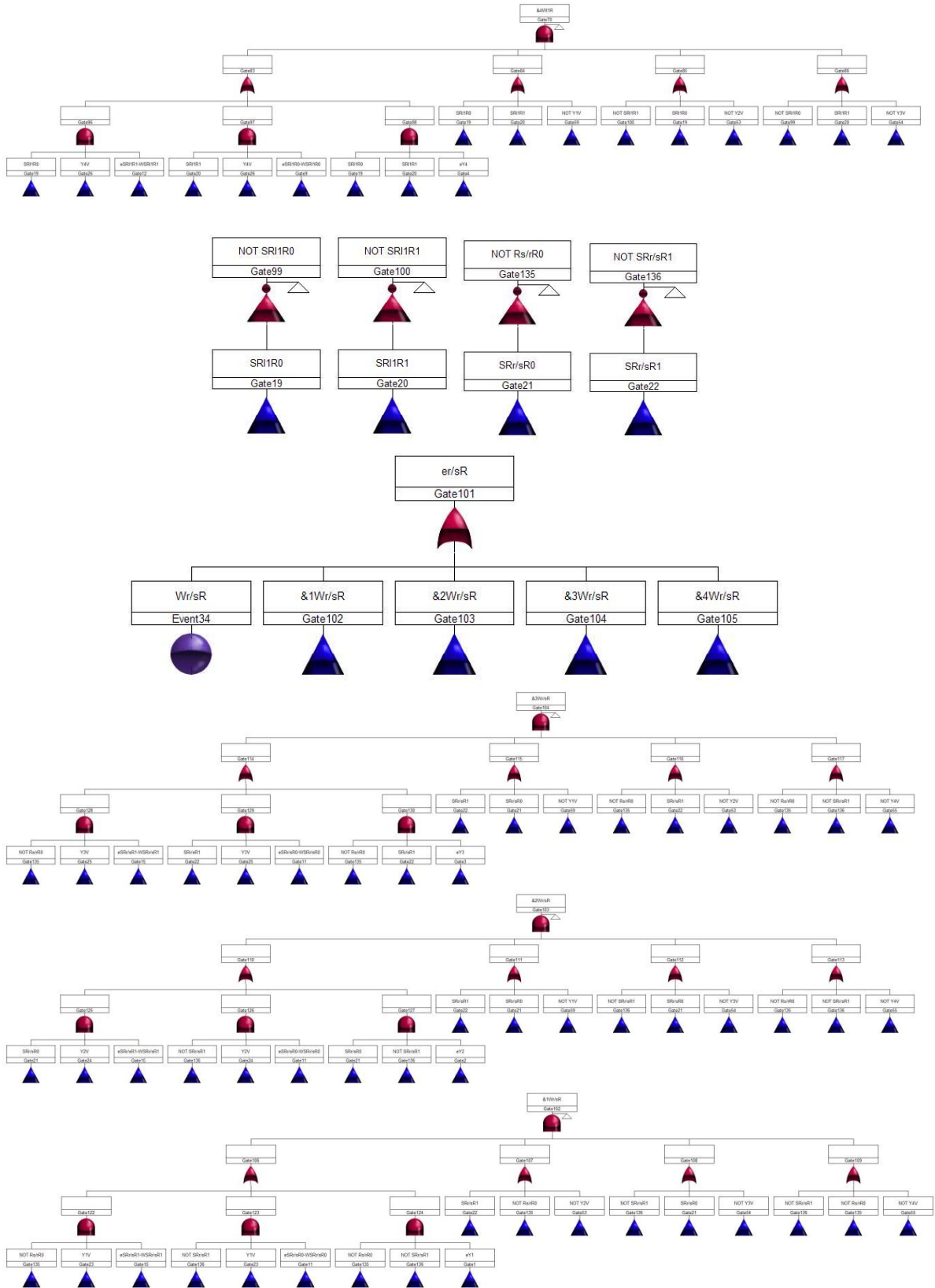


Fig.27.4 – Modelo de fallas del bloque de interconexión CB_1 genérico

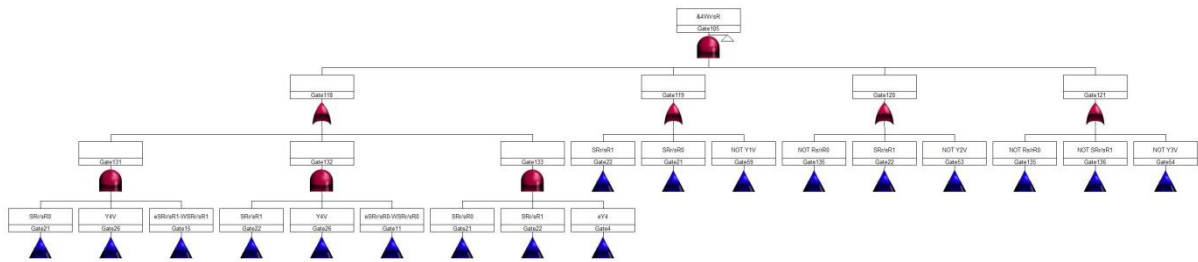


Fig.27.5 – Modelo de fallas del bloque de interconexión CB₁ genérico

6.6. Modelo de falla de un Bloque de conexión CB₂

En la Fig.28 se muestra el bloque genérico correspondiente al bloque colocado por debajo (down) de un CLB con las líneas horizontales de conexión interna.

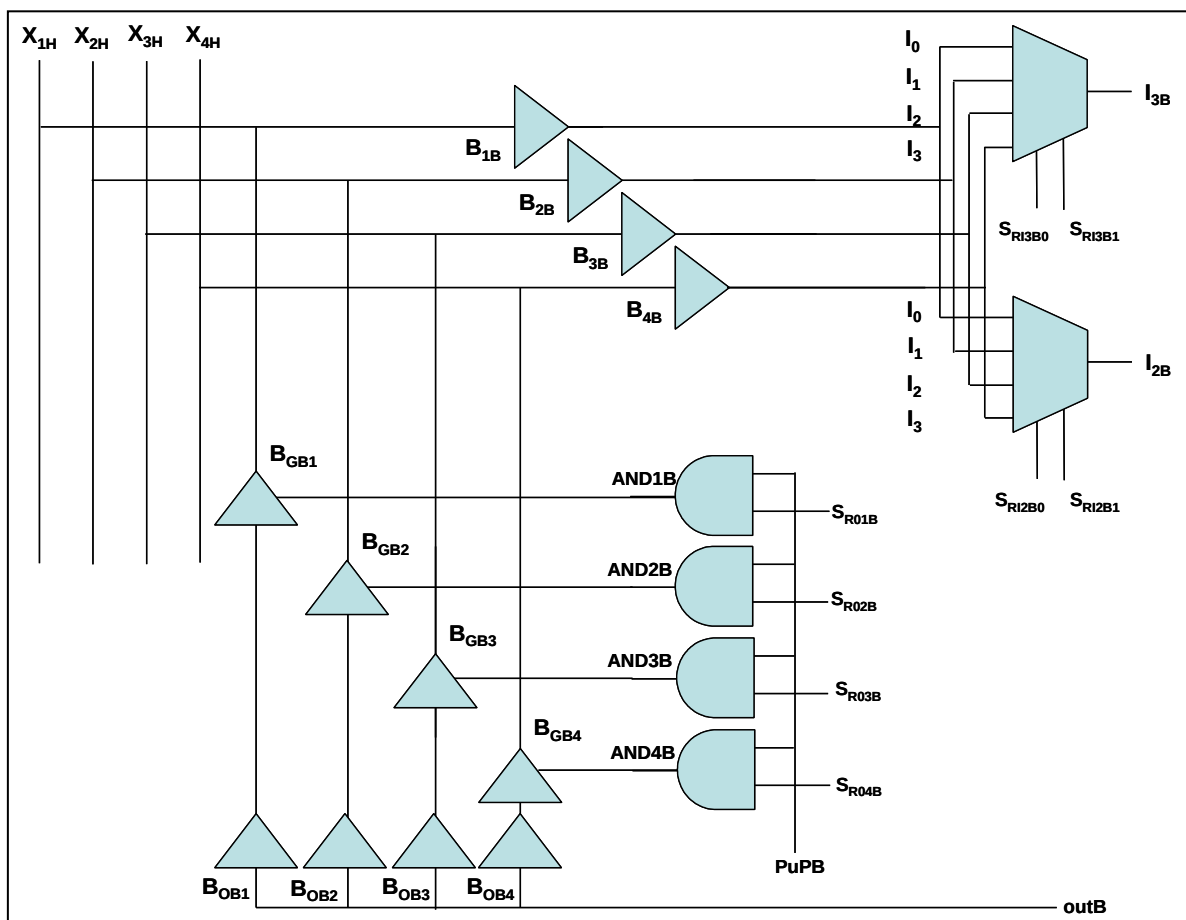


Fig.28 – Modelo del bloque de interconexión CB₂ genérico

El modelo de falla de un bloque CB_2 depende de las fallas intrínsecas de cada componente a saber:

Multiplexores = WI2B, WI3B

Buffers = B1B, B2B, B3B, B4B, BOB1, BOB2, BOB3, BOB4

Buffers Three State = BGB1, BGB2, BGB3, BGB4

Compuertas AND = AND1B, AND2B, AND3B, AND4B

Del estado de las entradas y biestable del mismo a saber:

CB_2 Entradas = SRI3B0, SRI3B1, SRI2B0, SRI2B1, X1H, X2H, X3H, X4H, outB, PuPB, SR01B, SR02B, SR03B, SR04B

y de los errores presentes en estas entradas y estados:

I/O Estados y Entradas = eSRI3B0, eSRI3B1, eSRI2B0, eSRI2B1, eoutB, ePuPB, eSR01B, eSR02B, eSR03B, eSR04B, eY1V, eY2V, eY3V, eY4V,

En las figuras 29.1 a 29.8 se muestran los árboles de falla correspondientes.

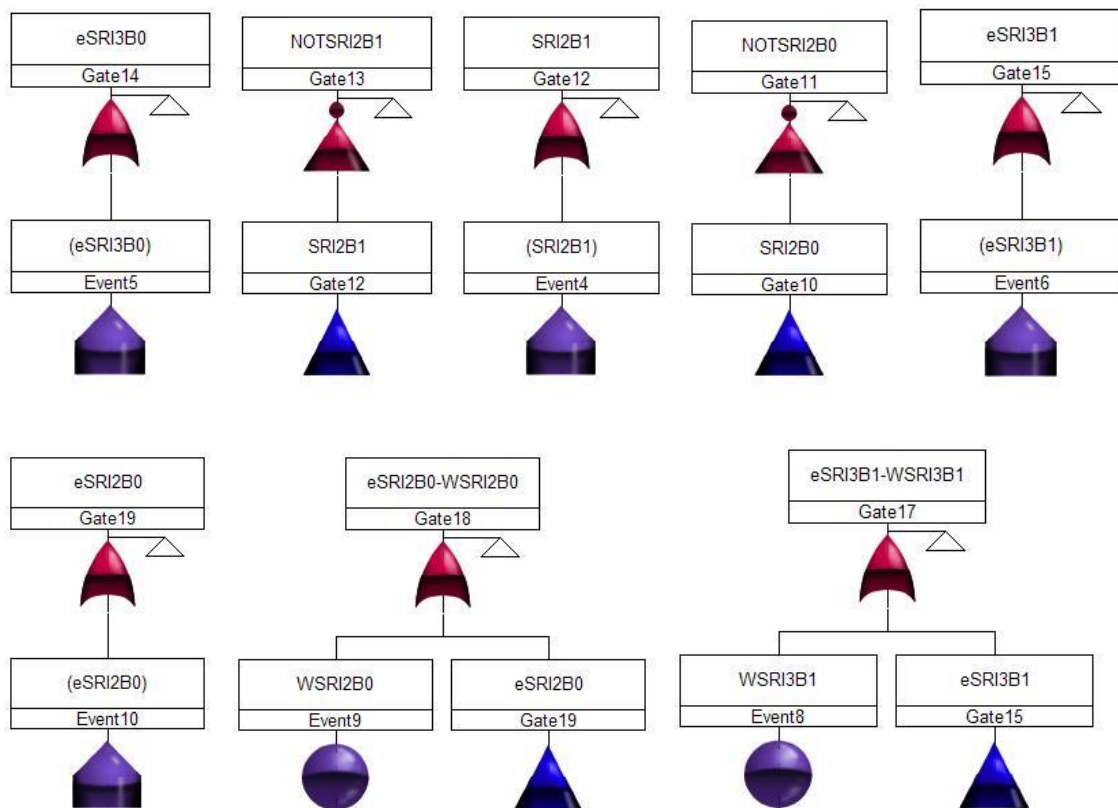


Fig.29.1 – Parcial del modelo de fallas del bloque de interconexión CB_2 genérico

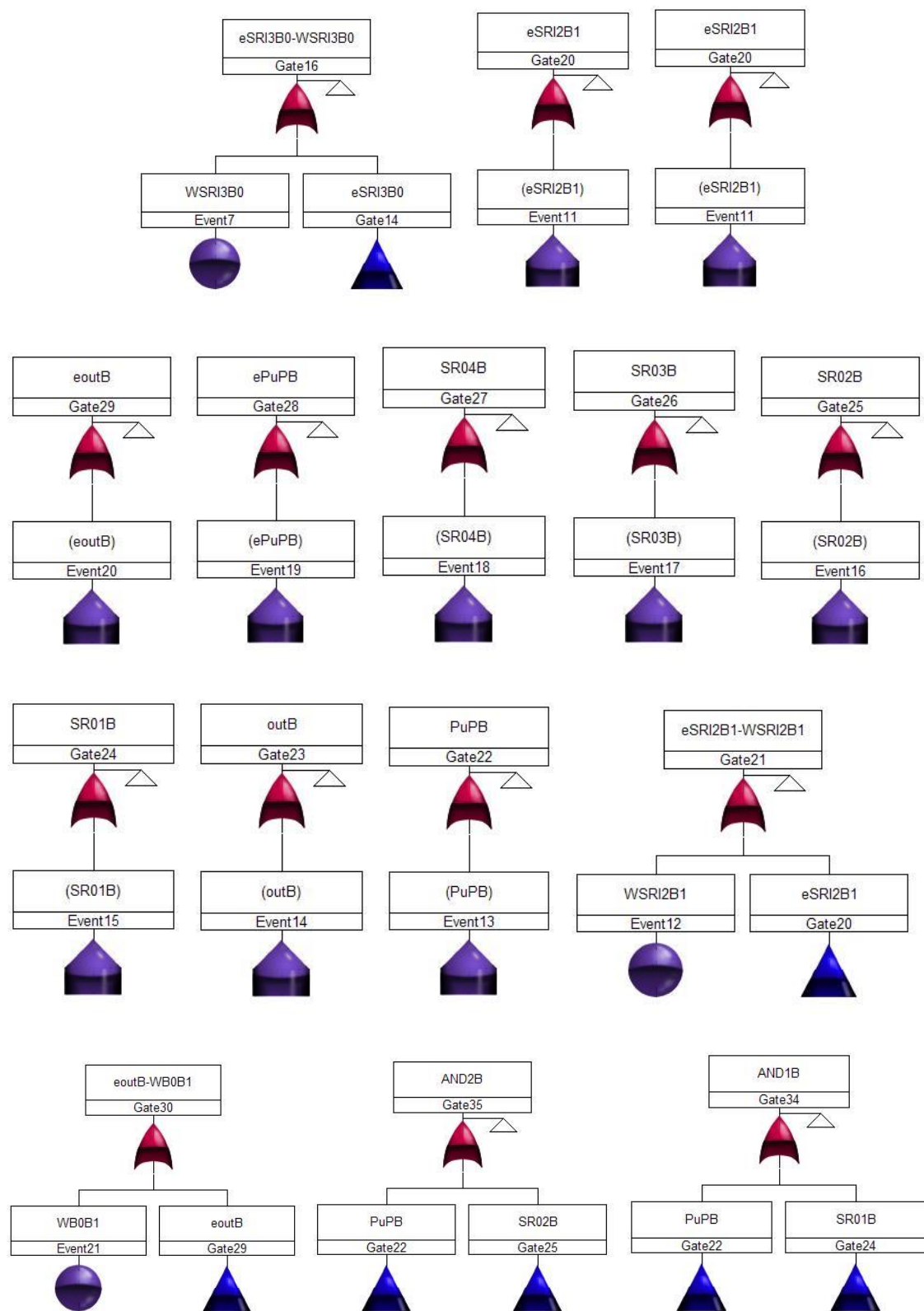


Fig.29.2 – Parcial del modelo de fallas del bloque de interconexión CB₂ genérico

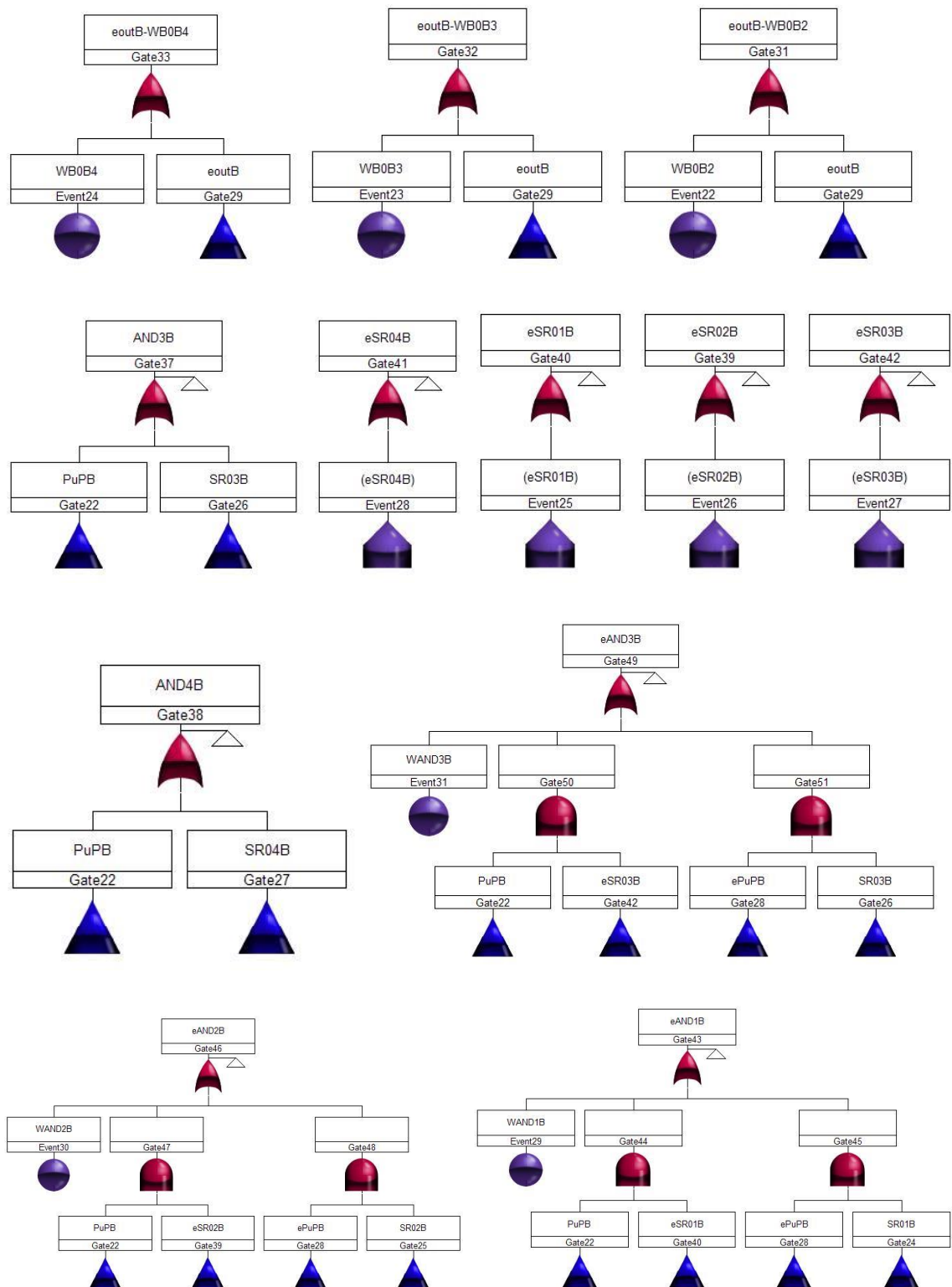


Fig.29.3 – Parcial del modelo de fallas del bloque de interconexión CB₂ genérico

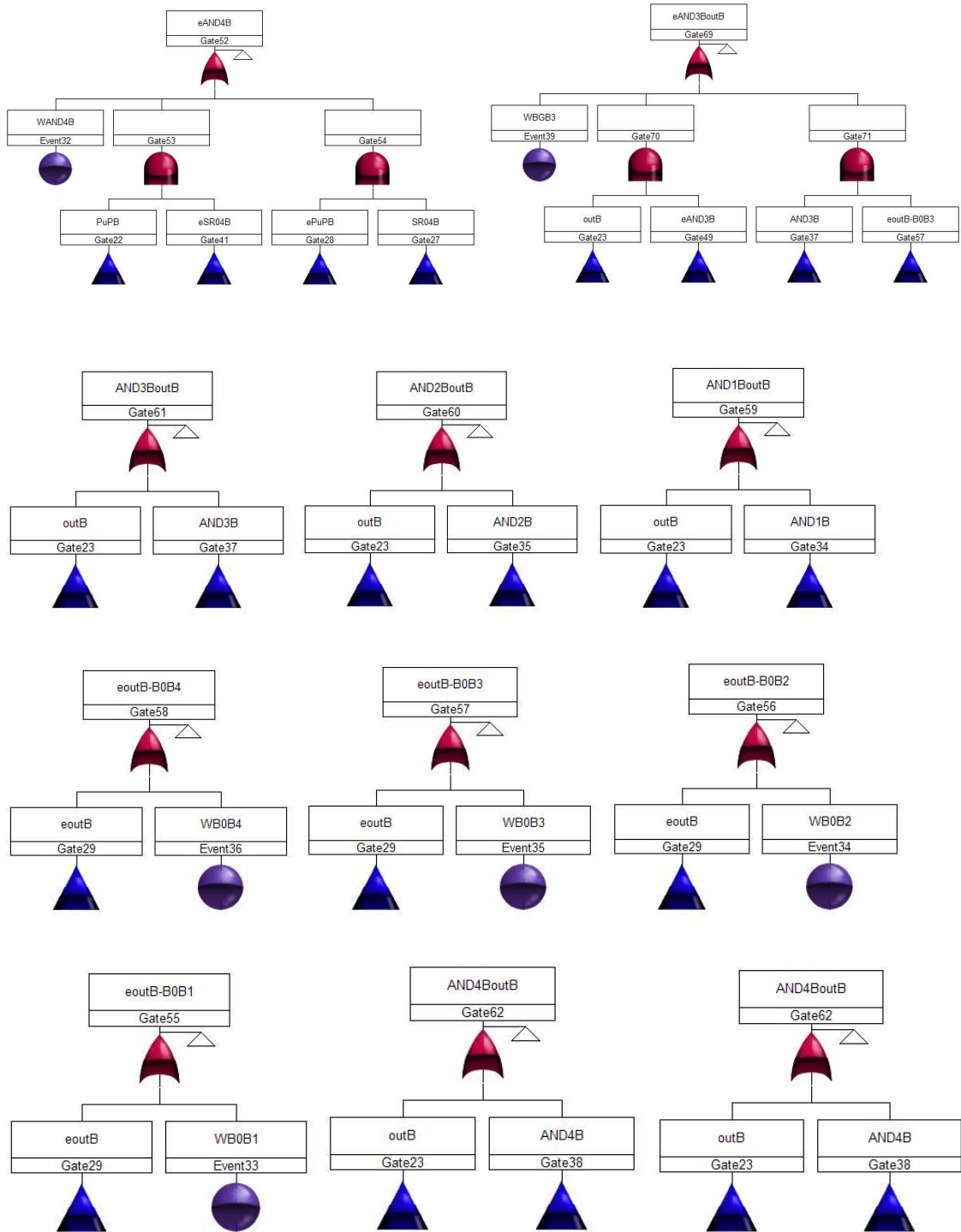


Fig.29.4 – Parcial del modelo de fallas del bloque de interconexión CB_2 genérico

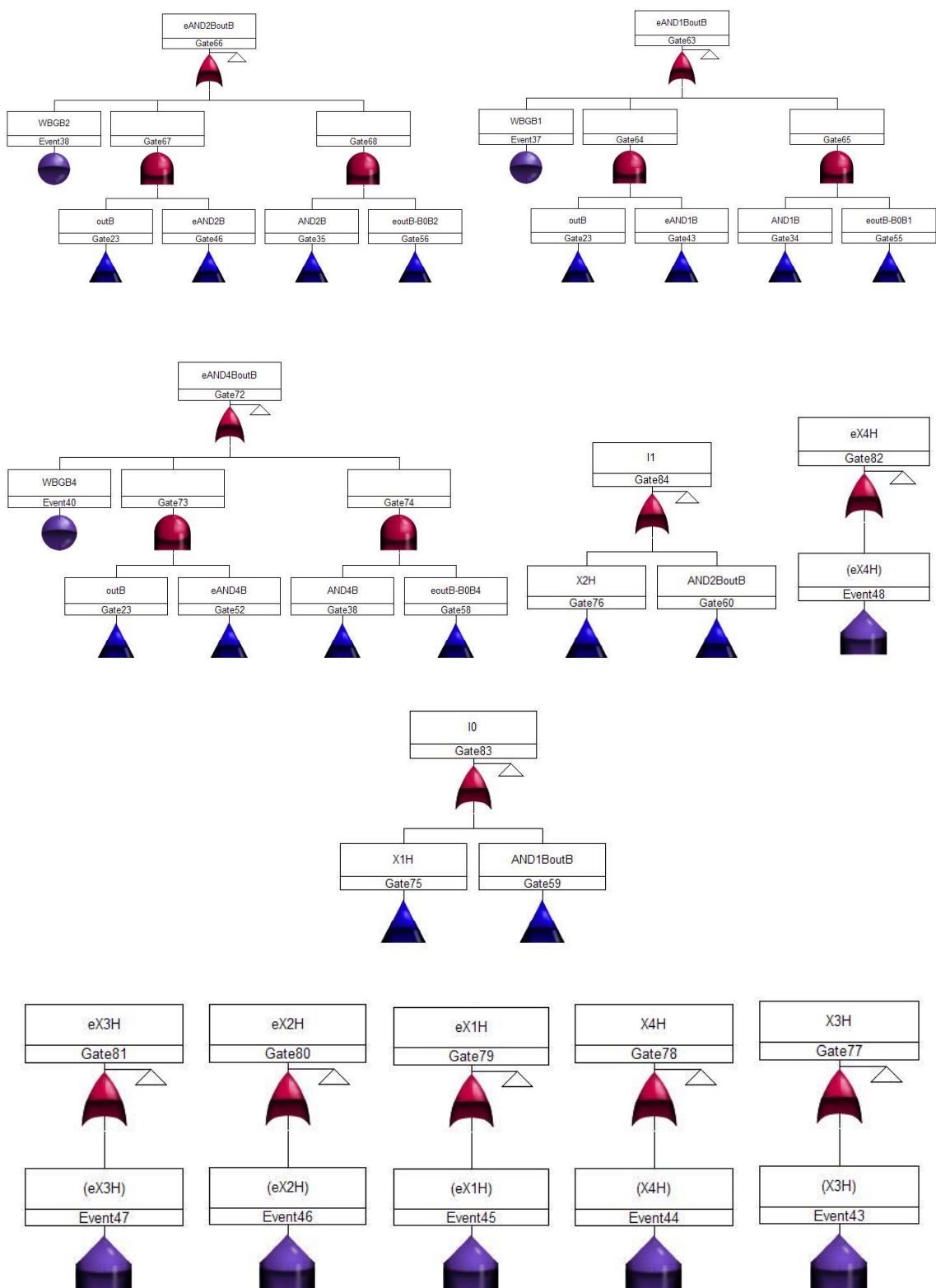
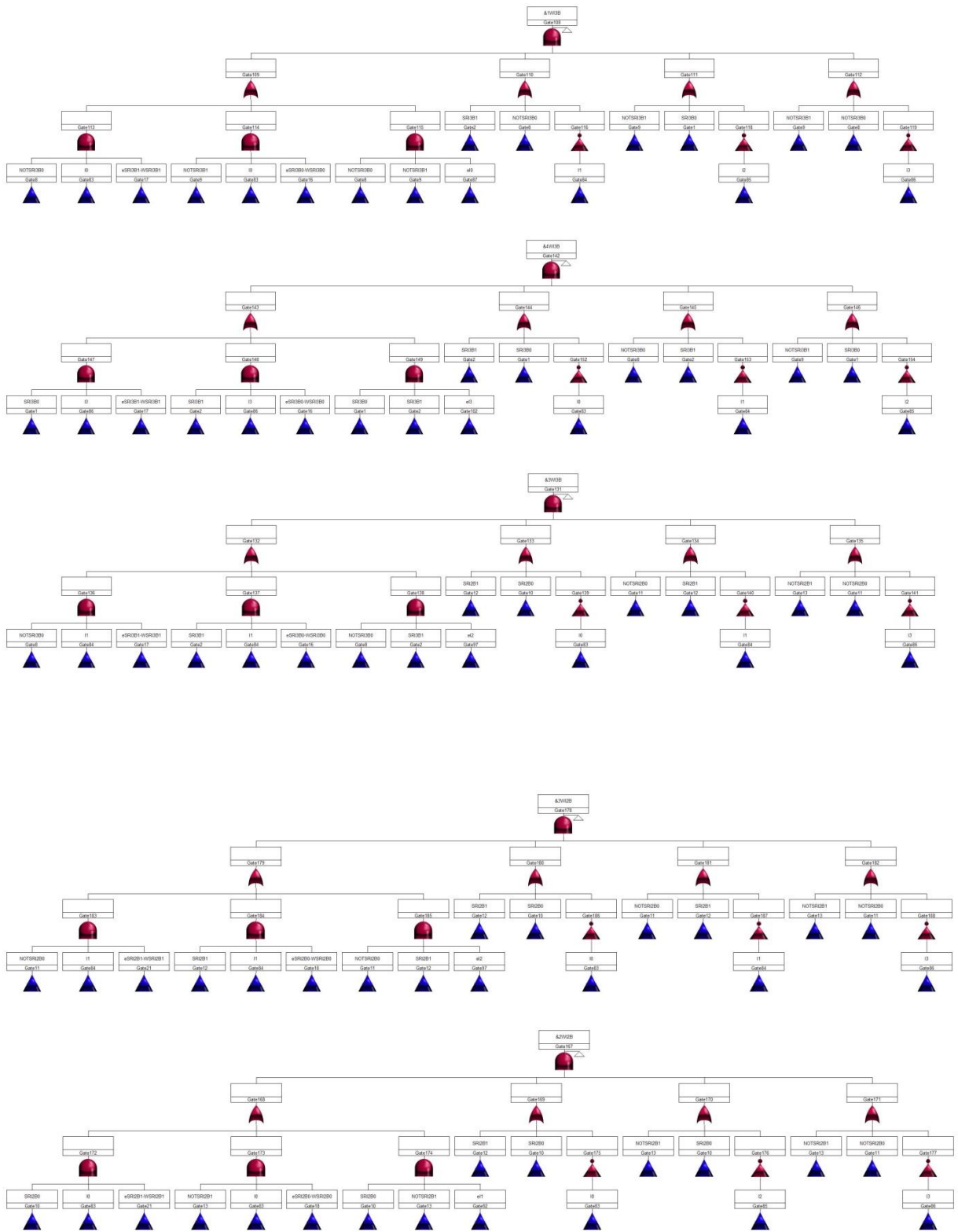


Fig.29.5 – Parcial del modelo de fallas del bloque de interconexión CB₂ genérico



F

Fig.29.7 – Parcial del modelo de fallas del bloque de interconexión CB₂ genérico

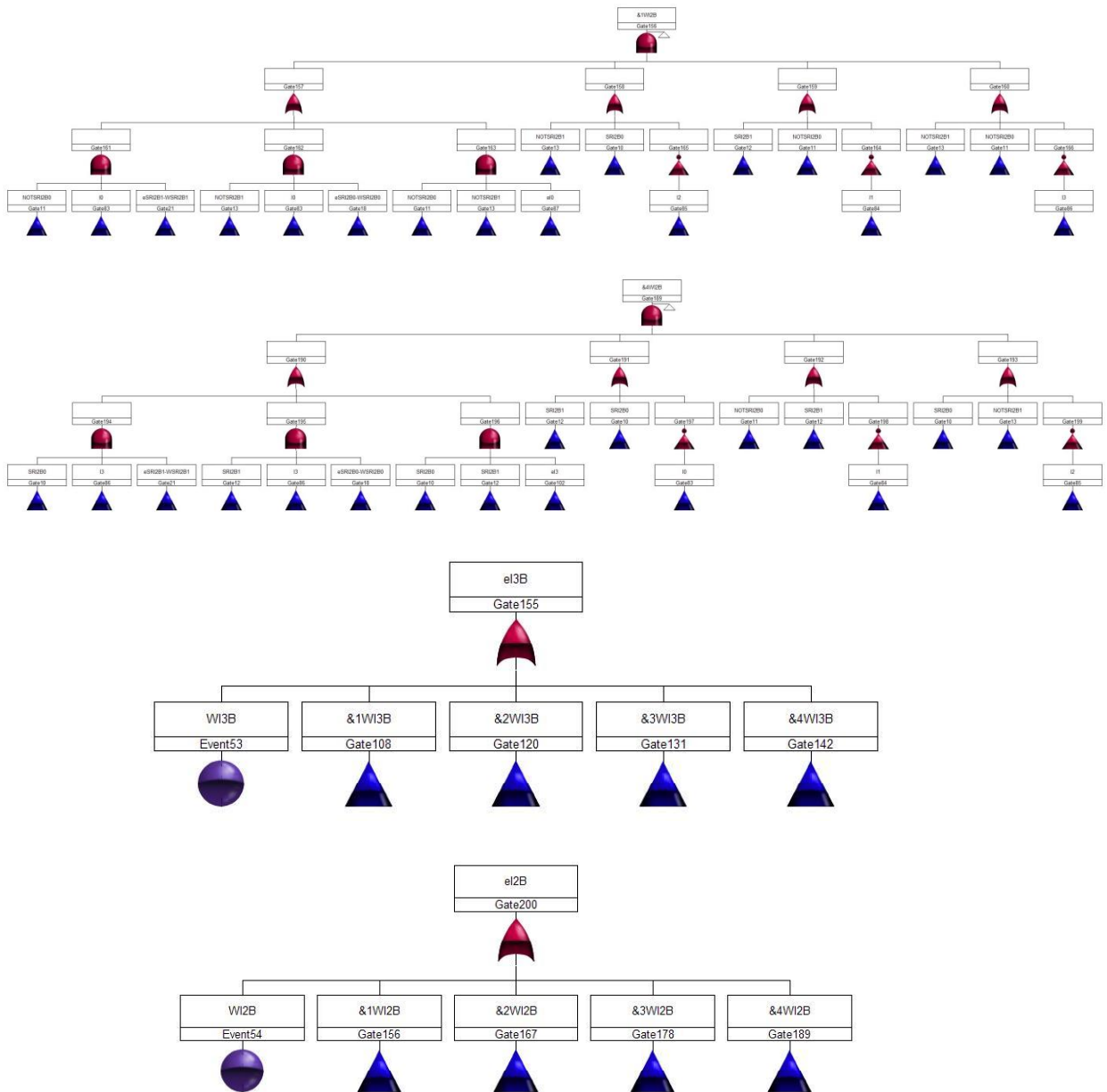


Fig.29.8 – Parcial del modelo de fallas del bloque de interconexión CB₂ genérico

7. Simulación y resultados

En la Fig. 30 se muestra una celda estándar de FPGA genérica. Se ha procedido a la interconexión de los diagramas correspondientes a los modelos de árboles de falla planteado para los bloques constitutivos de esta celda estándar. La celda esta constituida por un bloque CBL, dos CB₁, dos CB₂, 8 Input-Output y los SDR correspondientes al comando de los MUX respectivos.

Considerando errores presentes en las entradas y estados iniciales nulos y, por otro lado una tasa de fallas de cada unos de los componentes básicos constitutivos de los distintos bloques igual a $1.E-08 \text{ hs}^{-1}$ se ha procedido a correr modelo en el programa RELEX.

En las Fig. 31 se muestra el Setup del programa de simulación y es resultado obtenido para la probabilidad de fallas, en este caso tasa de fallas de la celda de FPGA genérica. El valor resultante es aproximadamente de $2.E-07 \text{ hs}^{-1}$.

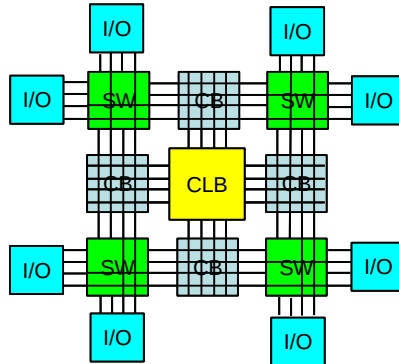


Fig.30 – Celda estándar de FPGA genérica

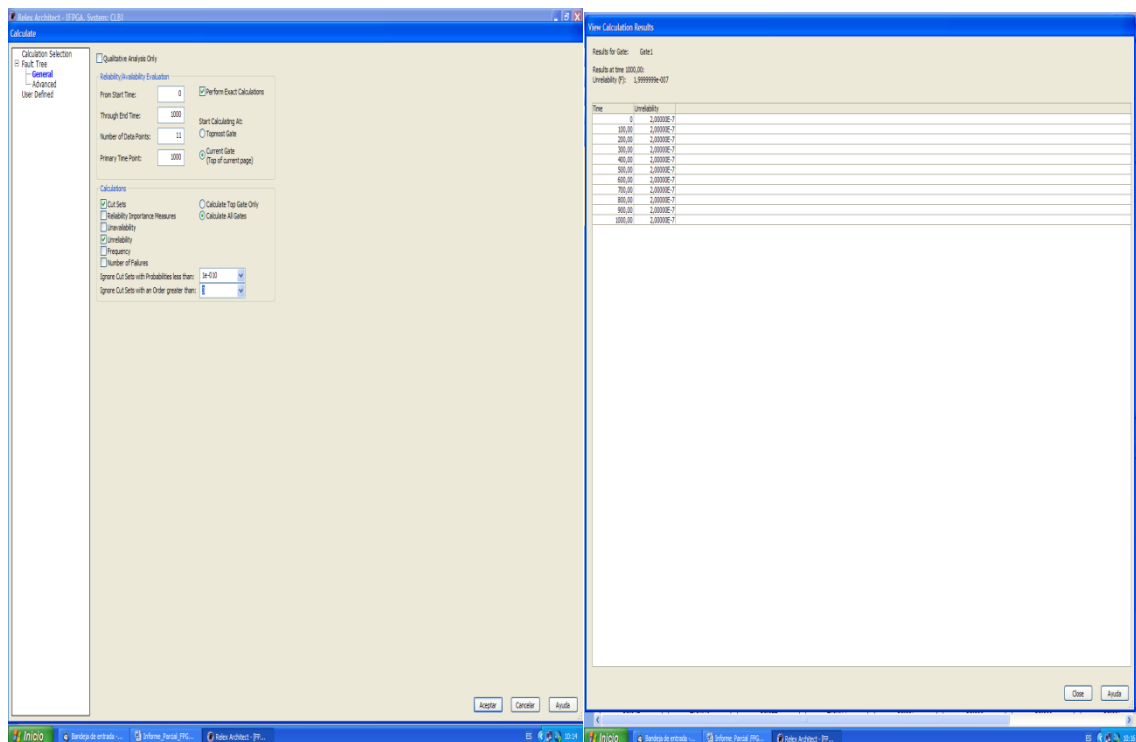


Fig.31 – Setup y resultados de la simulación en RELEX

Resulta de interés escalar este resultado a medida que se agregan celdas aumentando el área de aplicación de la FPGA. En la Fig. 32 se muestra este escalamiento. Se ha procedido a realizar las interconexiones correspondientes para cada caso hasta el número de 4 y su simulación en RELEX. En la Fig. 33 se muestra el conjunto de bloques del proyecto FPGA utilizado a esos efectos.

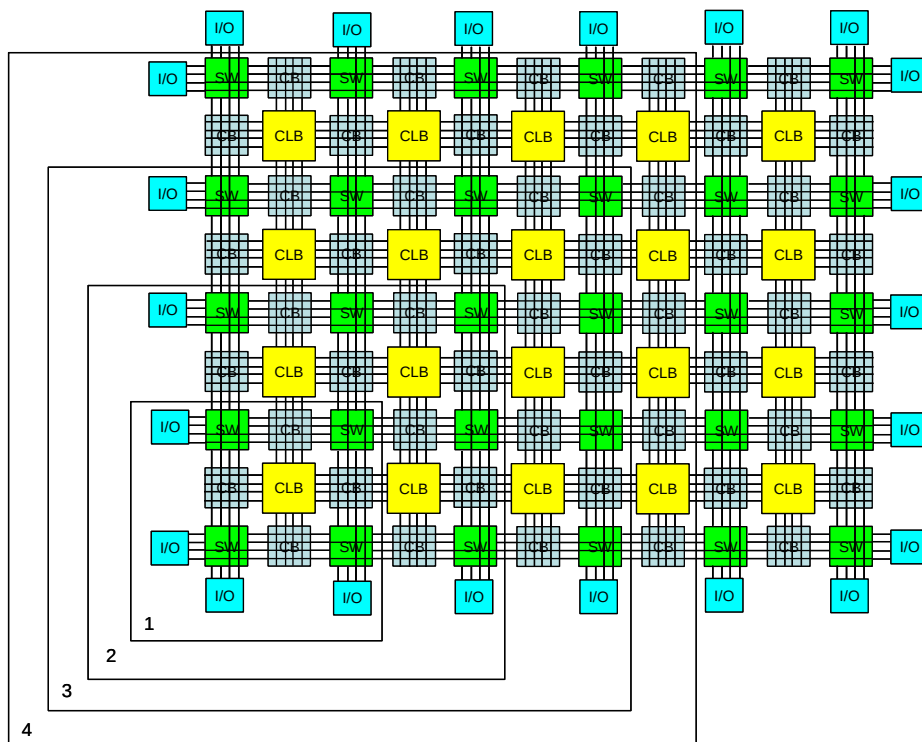


Fig.32 – Escalamiento de las áreas de aplicación

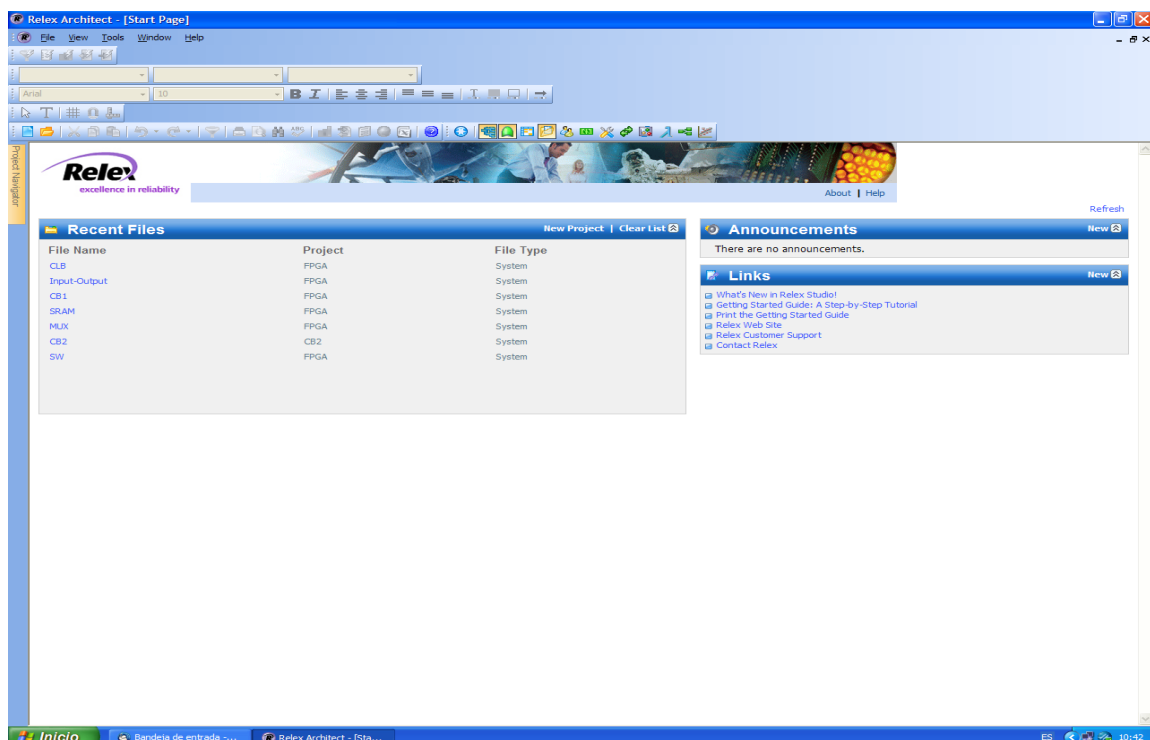


Fig.33 – Proyecto FPGA – Escalamiento de la simulación en RELEX

En el cuadro de la Fig.34 se muestra el resultado de estas simulaciones para 1, 2, 3 y 4 celdas y en la Fig.35 el grafico de valores de tasas de fallas versus número de celdas. La recta de tendencia es:

$$\text{Tasa de fallas} \approx 3.E-06 .(\# \text{ de celdas } -1)$$

Como se observa para una celda la tasa de fallas se hace casi nula y el MTBF extremadamente alto.

Escala	Tasa de fallas	MTBF
#	hs ⁻¹	hs
1	2,00E-07	5000000,00
2	3,00E-06	333333,33
3	7,00E-06	142857,14
4	1,00E-05	100000,00

Fig.34 – Número de celdas, Tasa de fallas y MTBF

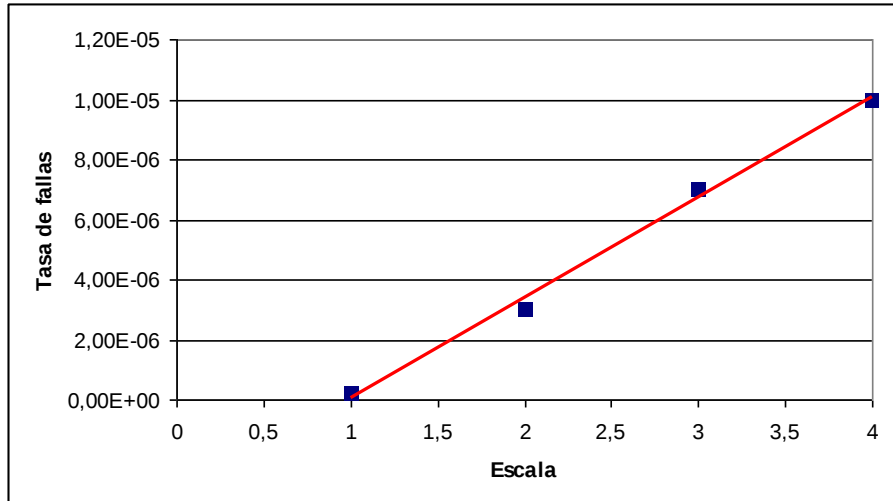


Fig.35 – Número de celdas versus Tasa de fallas

7. Conclusiones

Se ha cumplido totalmente con los puntos del programa presentado. La actuación del equipo de investigación ha sido el adecuado y ha estado a la altura de la complejidad del proyecto. La implementación del mismo vía el programa RELEX-Reliability Studio. PTC Company USA, propiedad de CONAE ha sido exitosa en todos los casos. La complejidad de los modelos a los que se ha arribado hace difícil su adaptación e implementación. Por otro lado la actuación de los integrantes del equipo de

investigación ha sido destacada. Se deja constancia además que el presupuesto asignado resulta coherente y satisfactorio por el momento, teniendo en cuenta el aporte de la Comisión Nacional de Actividades Espaciales en cuanto a recursos de tiempo y uso del programa RELEX citado precedentemente. Los resultados de esta investigación serán de gran utilidad en la evaluación de la confiabilidad de material de vuelo de la serie de satélites dentro del programa espacial que se encuentra en plena elaboración por los próximos 20 años.

8. Bibliografía

1. Reliability Evaluation of Logic Circuits, Microelectronics & Reliability, J.L.Roca, Vol. 25, N° 2, pp.257-260, Pergamon Press Ltd.
2. Architecture of Field-Programmable Gate Arrays, Jonathan Rose, Abbas El Gamal & Alberto Sangiovanni-Vincentelli, IEEE Proceedings, Vol.81, No.7, July 1993, pp. 1013-1029.
3. SRAM Based Re-programmable FPGA for Space Applications, J. J. Wang, R. B. Katz, J. S. Sun, B. E. Cronquist I., J. L. McCollum I, T. M. Speers I. & W. C. Plants, Actel Corporation, Sunnyvale, California 94086, NASA Goddard Space Flight Center, Greenbelt, Maryland 20771.
4. Reconfigurable Field Programmable Gate Arrays for Mission-Critical Applications, N. Battezzati et al., DOI 10.1007/978-1-4419-7595-9_2, Springer Science+Business Media, LLC 2011, pp.7-35.
5. Application-Specific Mesh-based Heterogeneous FPGA Architectures, H. Parvez and H. Mehrez, DOI 10.1007/978-1-4419-7928-5_2, Springer Science+Business Media, LLC 2011, pp.9-30.
6. U. Farooq et al., Tree-Based Heterogeneous FPGA Architectures, U. Farooq et al., DOI: 10.1007/978-1-4614-3594-5_2, Springer Science+Business Media New York 2012, pp. 7-48.
7. A Tutorial on FPGA Routing, Daniel Gomez-Prado Maciej Ciesielski, Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, USA, pp.1-16.
8. FPGA Routing Structures: A Novel Switch Block And Depopulated Interconnect Matrix Architectures By Muhammad Imran Masud B.Eng., The Dalhousie University, 1998, pp.1-70.
9. Full Custom Layout of an SRAM-Based FPGA Blair Fort, Daniele Paladino, Franjo Plavec, ECE1388, University Of Toronto, December 2004