

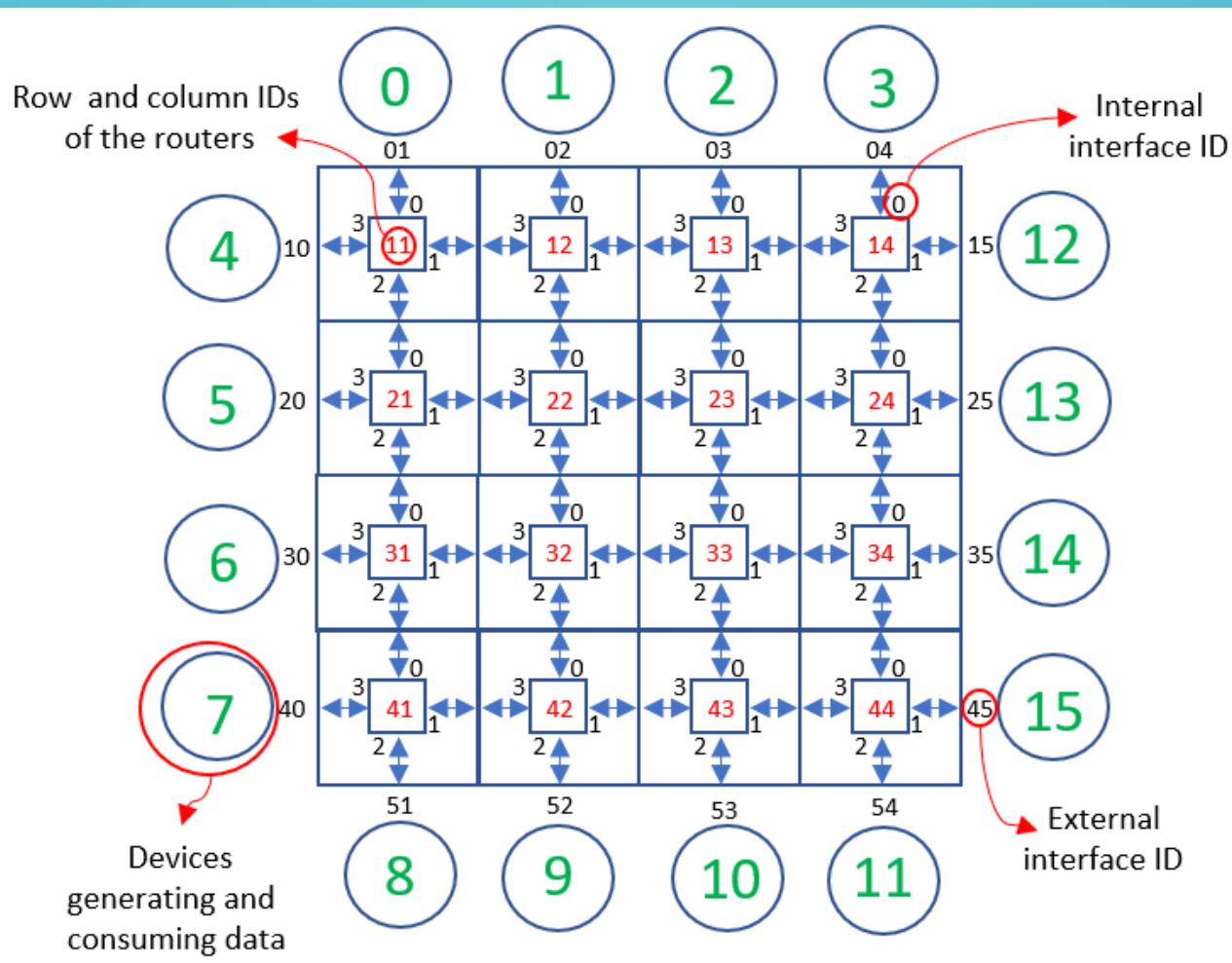
# PROYECTO 2

EN GRUPOS DE 2

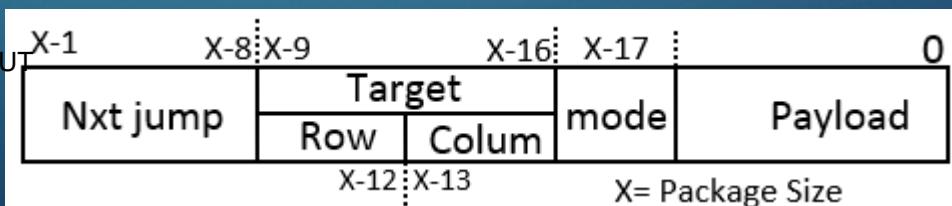
ENTREGA: SÁBADO 22 DE NOVIEMBRE 2025

## REQUISITOS MÍNIMOS PARA REVISAR EL PROYECTO.

- El proyecto debe compilar y correr.
- No debe haber plagio.
- Cada grupo realizará una demostración del trabajo.

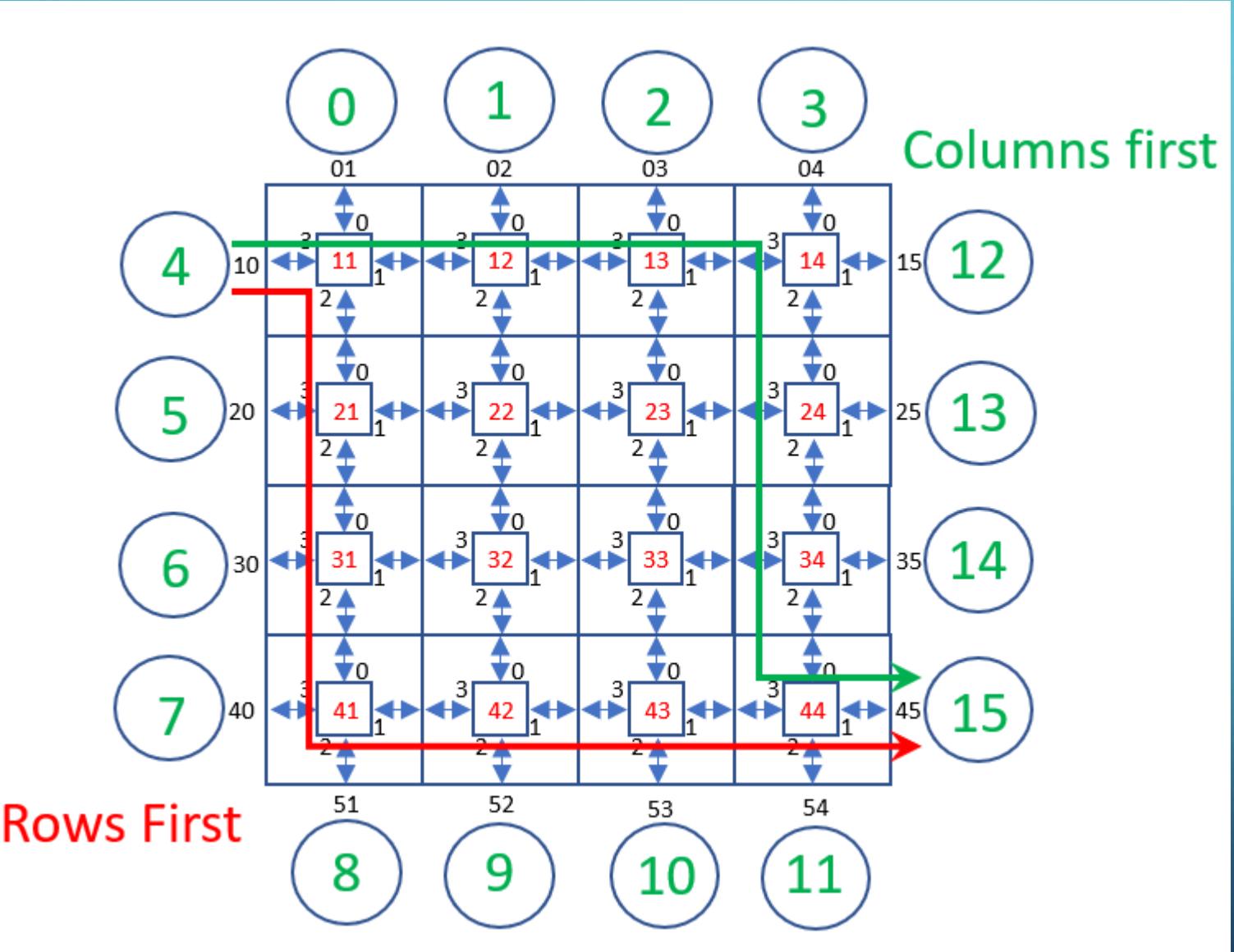


Nxt jump se genera automáticamente en el DUT



Bit mode: Indica si se quiere hacer fila primero o columna primero.

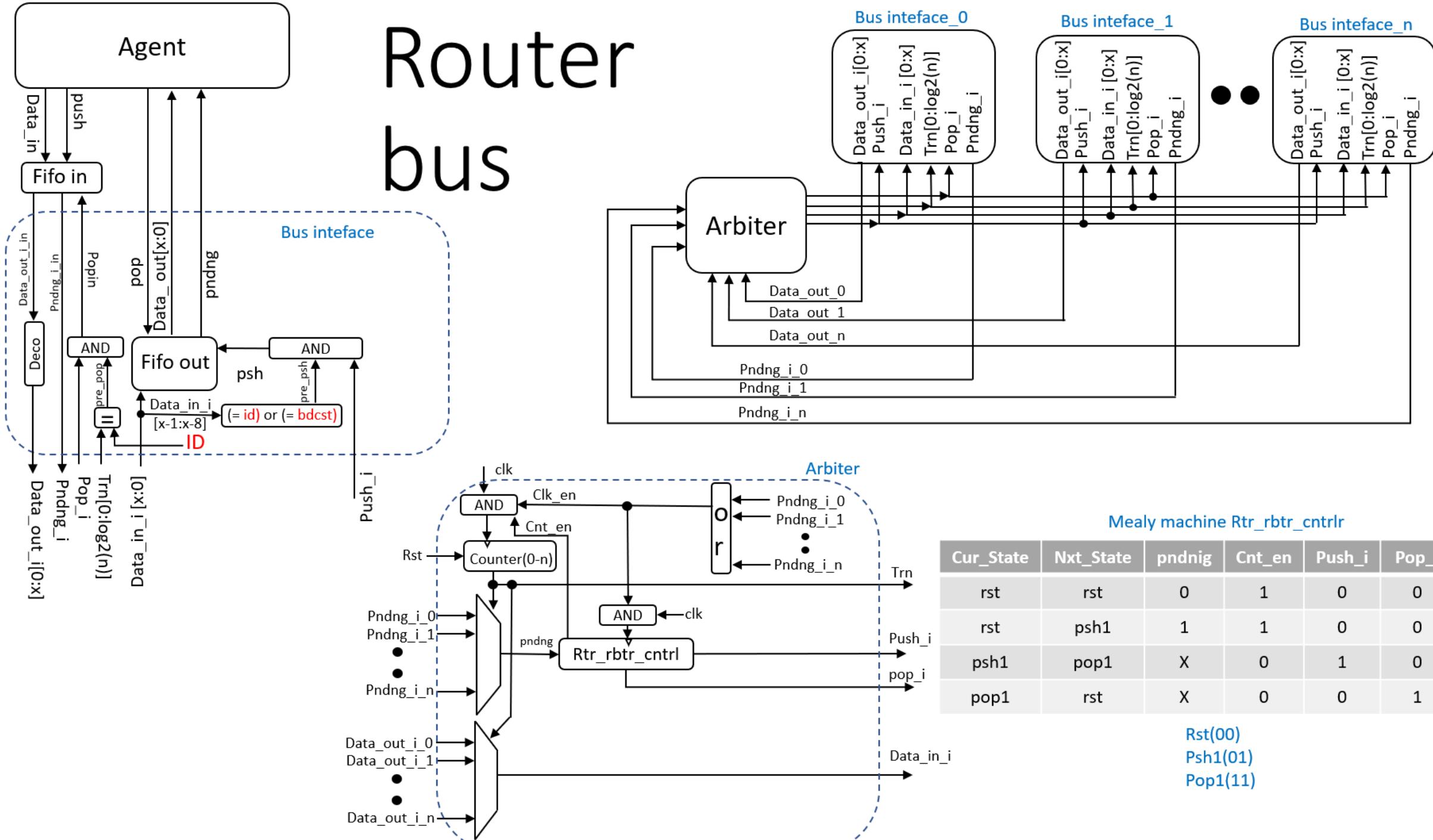
Payload:

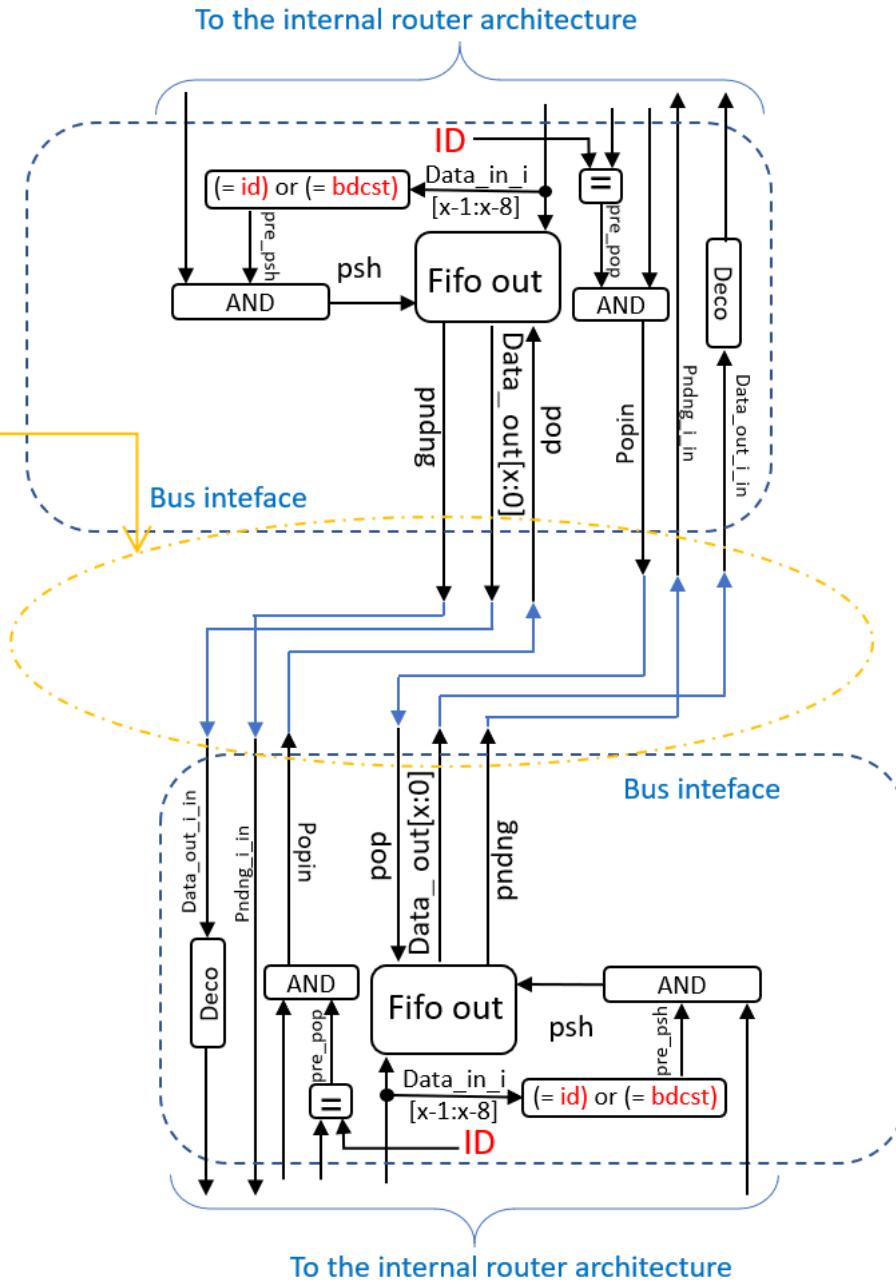
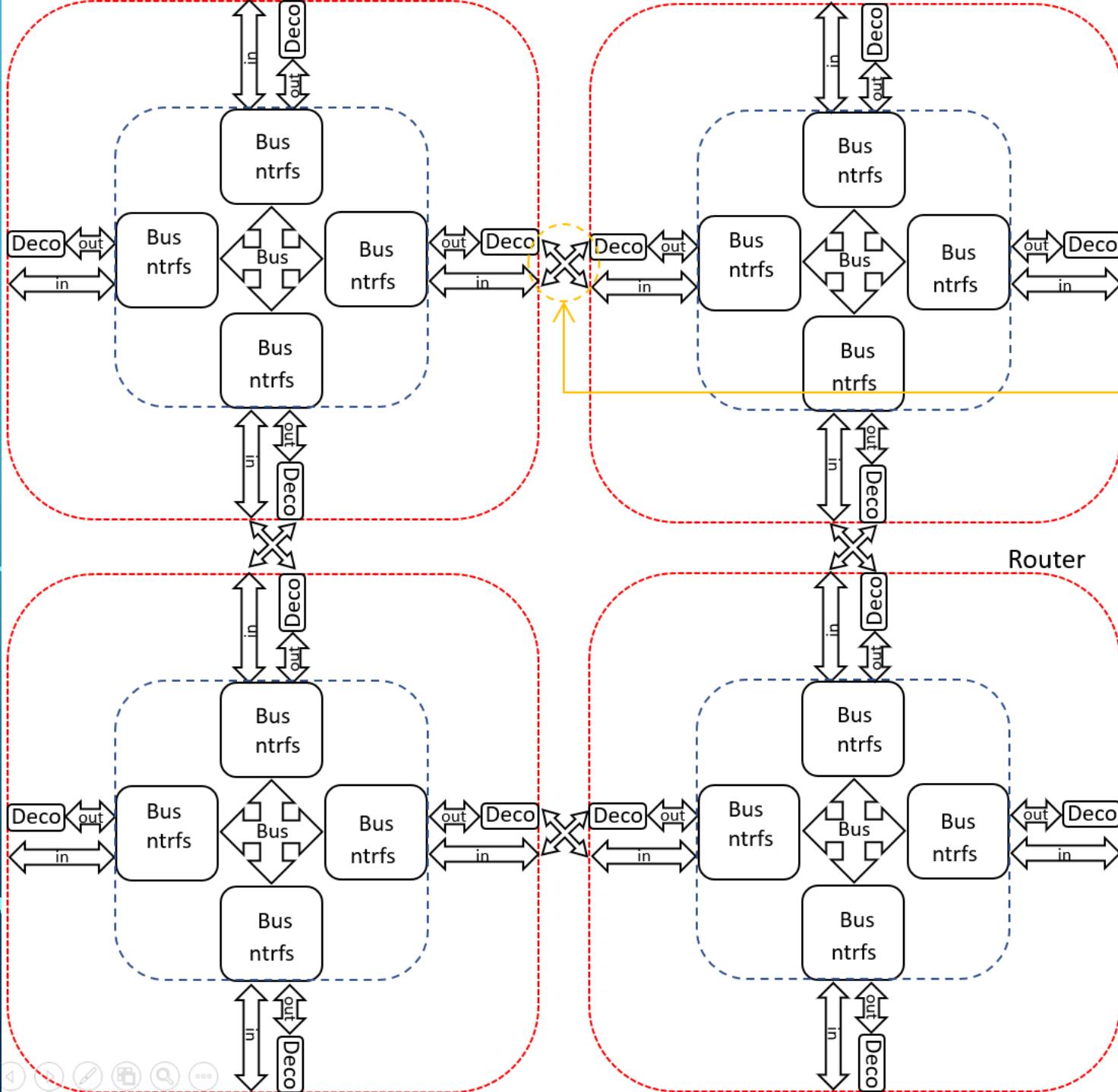


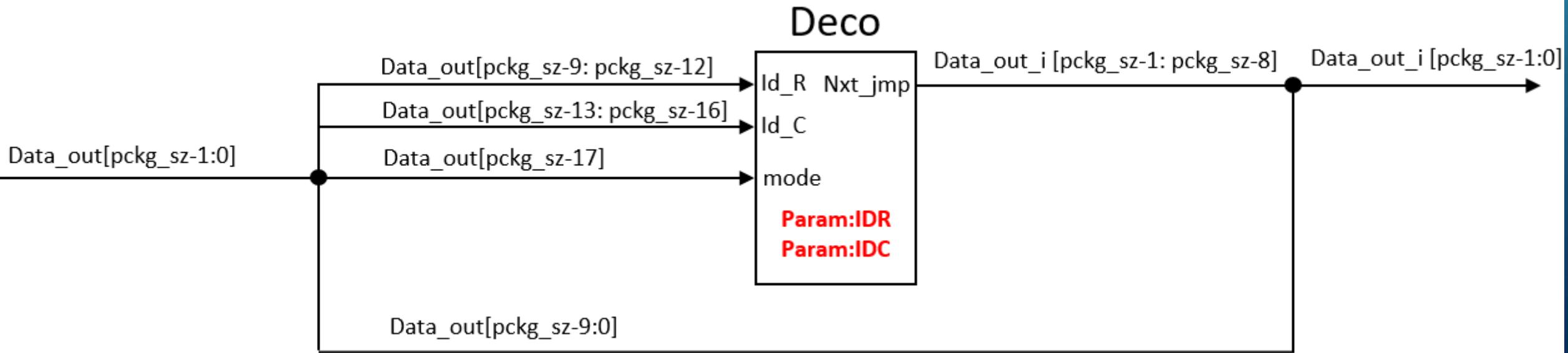
Si las FIFO que tengo para realizar las comunicaciones entre los distintos router, se llenan, voy a empezar a perder datos. Esto no está mal en sí, pero sí hay que tenerlo en cuenta, para cuando se realice la golden reference, está tome eso en cuenta y se comporte bien.

- Si el modo del paquete es 1 entonces enruta primero fila.

# Router bus







# PROTOTIPO DEL DUT

```
mesh_gnrtr #( .ROWS(`ROWS), .COLUMNS(`COLUMNS), .pckg_sz(`PCKG_SZ), .fifo_depth(`FIFO_DEPTH), .bdcst(`BROADCAST) ) uut(  
    .clk(clk),  
    .reset(reset),  
    .pndng(pndng),  
    .data_out(data_out),  
    .popin(popin),  
    .pop(pop),  
    .data_out_i_in(data_out_i_in),  
    .pndng_i_in(pndng_i_in)  
);
```

# EN EL CÓDIGO DEL ROUTER:

ESTO ES HARDWARE, está ligado  
al tiempo

```
 `ifdef DEBUG
  always@(posedge pop) begin
    $display("ntrfs: Message send in terminal: %g router ID: %g %g at time %g", ntrfs_id,id_r,id_c,$time );
    $display("trgt_r: %g", data_out[pck_sz-9:pck_sz-12]);
    $display("trgt_c: %g", data_out[pck_sz-13:pck_sz-16]);
    $display("mode: %g",data_out[pck_sz-17]);
    $display("src: %g",data_out[pck_sz-18:pck_sz-25]);
    $display("id: %g",data_out[pck_sz-26:pck_sz-33]);
    $display("pyld: %h",data_out[pck_sz-34:0]);
  end
  always@(posedge popin) begin
    $display("ntrfs: Message received in terminal: %g router ID: %g %g at time %g", ntrfs_id,id_r,id_c,$time );
    $display("trgt_r: %g", data_out_i_in[pck_sz-9:pck_sz-12]);
    $display("trgt_c: %g", data_out_i_in[pck_sz-13:pck_sz-16]);
    $display("mode: %g",data_out_i_in[pck_sz-17]);
    $display("src: %g",data_out_i_in[pck_sz-18:pck_sz-25]);
    $display("id: %g",data_out_i_in[pck_sz-26:pck_sz-33]);
    $display("pyld: %h",data_out_i_in[pck_sz-34:0]);
  end
`endif
```

# ARCHIVOS BASE:

```
13 `define FIFO
14 `include "fifo.sv"
15 `include "Library.sv"
16 `define LIB
17 `include "Router_library.sv"
```

- Las bibliotecas base con la implementación del DUT se encuentran en el TECDigital.

# ENTREGABLES:

- Test plan de todas las capacidades del diseño. (10%)
- Todas las unidades del ambiente de pruebas aleatorizadas controladas vistas en clase. Cada unidad debe correr en un proceso independiente. (10%) UVM
- Código con comentarios. (10%)
- Diseño con inclusión de aserciones en todas las interfaces principales.(10%)
- Cobertura de código y de casos mayor o igual al 95% incluyendo todos los estados validos posibles de los registros. (10%)
- El diseño de las pruebas debe incluir capacidad de aleatorización de: (20%)
  - Número de transacciones x terminal
  - Tiempos de envío de mensajes.
  - Mensajes con errores.Mandarlo a una salida que no existe. Verificar en los routers si el paquete sí corresponde para esta entrada, si no iba para el router que lo recibió, debe generar un error y borrarlo
  - Direcciones de destino de los mensajes
  - Identificador de “broadcast” QUITAR! :)

# ENTREGABLES:

No vamos a aleatorizar la cantidad de dispositivos, ni la profundidad de las FIFOs

- Identificación e implementación de casos de esquina. (10%)
- El ambiente de pruebas debe ser capaz de generar datos de: (20%)
  - a. Retraso promedio en la entrega de paquetes x terminal y general en función de la cantidad de dispositivos y las profundidad de las FIFOs.
  - b. Ancho de banda promedio máximo y mínimo, en función de la cantidad de dispositivos y las profundidad de las FIFOs.
  - c. Debe ser capaz de entregar un reporte de los paquetes enviados recibidos en formato csv. Se debe incluir tiempo de envío terminal de procedencia, terminal de destino tiempo de recibido, retraso en el envío.
  - d. Usando GNUpot pueden usar los resultados del reporte anterior y generar los gráficos que se solicitan en a y b.