GDS Green and Dependable Systems Laboratory
Computing Systems Laboratories

myNetListParser

M1 Shunsuke Yoshihara

Structure of module

- struct _module {
  - int nnl; /* Number of Net */
  - int npi; /* Number of Primary Input */
  - int npo; /* Number of Primary Output */
  - int nff; /* Number of Flip-Flop */

  - netlist *netlistArr [MAX_NETLIST]; /* Pointer Array of netlist */
  - netlist *ffl [MAX_NETLIST]; /* Pointer Array of Flip-Flop List */
  - netlist *pil [MAX_NETLIST]; /* Pointer Array of PI List */
  - netlist *pol [MAX_NETLIST]; /* Pointer Array of PO List */
- } module;

Structure of Netlist

- struct _netlist {
  - int num; /* Net Number (ID) */
  - int type; /* Net Type */
  - int nfi; /* Number of Fanin */
  - int nfo; /* Number of Fanout */

  - port **fin; /* Pointer to Fanin Port (require malloc) */
  - port **fout; /* Pointer to Fanout Port (require malloc) */

  - char *instName; /* Instance Name */

- } netlist;

Structure of Port

- struct _port {
  - netlist *nl; /* Netlist with this port */
  - char *portName; /* Port Name (ex “.D”, “.Q “) */
  - char *netName; /* Net(Signal) Name (ex “n7”) */
  - port **connNext; /* Next Port (Fout -> Fin) */
  - int connCount; /* Number of Next Port */
- } port;

Relationship

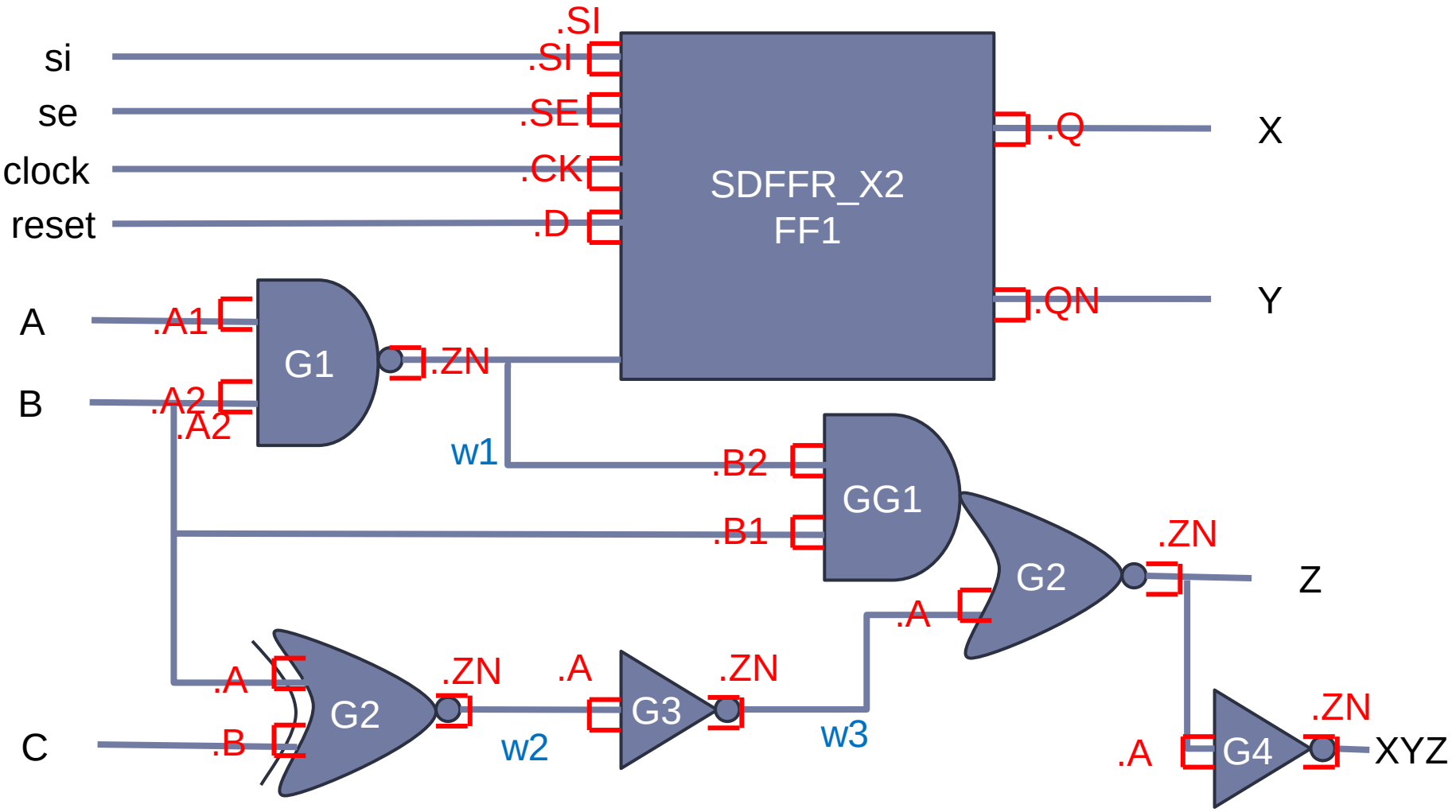
- module
  - netlist[0]
    - finport[0],…
    - foutport[0],…
  - netlist[1]
    - finport[0],…
    - foutport[0],…
  - netlist[2]
    - finport[0],…
    - foutport[0],…
- port
  - netlist
  - portName
  - netName
  - Connect Count
  - Next Port or Prev Port[0],
    Next Port or Prev Port[1],
    Next Port or Prev Port[2],…

Sample Circuit (_test2.v)

- input A, B, C, clock, si, se, reset;
- output X, Y, Z, XYZ;
- wire w1, w2, w3;

- SDFFR_X2 FF1 (.D(w1), .SI(si), .SE(se), .CK(clock), .RN(
- reset), .Q(X), .QN(Y));
- NAND2_X1 G1 (.A1(A), .A2(B), .ZN(w1));
- XNOR2_X1 G2 (.A(B), .B(C), .ZN(w2));
- INV_X1 G3 (.A(w2), .ZN(w3));
- AOI21_X1 GG1 (.B1(B), .B2(w1), .A(w3), .ZN(Z));
- INV_X1 G4 (.A(Z), .ZN(XYZ));

Schema of Sample Circuit(_test2.v)



./saa –n circuit/_test2.v

```
[v25y2019@cpu myNetListParser]$ cat saa.txt
=== Module info ===
 netlist count = 20
 npi=7, npo=4, nff=1
--------------------------------------------------------------------------------
 [signal  #0] type=20, nfi=0, nfo=1, signal=A
   ↓
   --------> Next Port[0]:  instance = G1, port = .A1, net = A

--------------------------------------------------------------------------------
 [signal  #1] type=20, nfi=0, nfo=1, signal=B
   ↓
   --------> Next Port[0]:  instance = G1, port = .A2, net = B
   --------> Next Port[1]:  instance = G2, port = .A, net = B
   --------> Next Port[2]:  instance = GG1, port = .B1, net = B

--------------------------------------------------------------------------------
 [signal  #2] type=20, nfi=0, nfo=1, signal=C
   ↓
   --------> Next Port[0]:  instance = G2, port = .B, net = C

--------------------------------------------------------------------------------
 [signal  #3] type=20, nfi=0, nfo=1, signal=clock
   ↓
   --------> Next Port[0]:  instance = FF1, port = .CK, net = clock

--------------------------------------------------------------------------------
 [signal  #4] type=20, nfi=0, nfo=1, signal=si
   ↓
   --------> Next Port[0]:  instance = FF1, port = .SI, net = si

--------------------------------------------------------------------------------
 [signal  #5] type=20, nfi=0, nfo=1, signal=se
   ↓
   --------> Next Port[0]:  instance = FF1, port = .SE, net = se

--------------------------------------------------------------------------------
 [signal  #6] type=20, nfi=0, nfo=1, signal=reset
   ↓
   --------> Next Port[0]:  instance = FF1, port = .RN, net = reset

--------------------------------------------------------------------------------
```

```
--------------------------------------------------------------------------------
 [netlist #17] type=4, nfi=1, nfo=1, inst=G3
   ###fanin[0] of [netlist #17]: port=.A, signal=w2, inst=G3
   ↓
   --------> Prev Port[0]:  instance = G2, port = .ZN, net = w2

   ###fanout[0] of [netlist #17]: port=.ZN, signal=w3, inst=G3
   ↓
   --------> Next Port[0]:  instance = GG1, port = .A, net = w3

--------------------------------------------------------------------------------
 [netlist #18] type=8, nfi=3, nfo=1, inst=GG1
   ###fanin[0] of [netlist #18]: port=.B1, signal=B, inst=GG1
   ↓
   --------> Prev Port[0]:  instance = B, port = --, net = B

   ###fanin[1] of [netlist #18]: port=.B2, signal=w1, inst=GG1
   ↓
   --------> Prev Port[0]:  instance = G1, port = .ZN, net = w1

   ###fanin[2] of [netlist #18]: port=.A, signal=w3, inst=GG1
   ↓
   --------> Prev Port[0]:  instance = G3, port = .ZN, net = w3

   ###fanout[0] of [netlist #18]: port=.ZN, signal=Z, inst=GG1
   ↓
   --------> Next Port[0]:  instance = G4, port = .A, net = Z
   --------> Next Port[1]:  instance = Z, port = --, net = Z

--------------------------------------------------------------------------------
 [netlist #19] type=4, nfi=1, nfo=1, inst=G4
   ###fanin[0] of [netlist #19]: port=.A, signal=Z, inst=G4
   ↓
   --------> Prev Port[0]:  instance = GG1, port = .ZN, net = Z

   ###fanout[0] of [netlist #19]: port=.ZN, signal=XYZ, inst=G4
   ↓
   --------> Next Port[0]:  instance = XYZ, port = --, net = XYZ

=== End of module info ===
[v25y2019@cpu myNetListParser]$
```

構造体と接続関係の構築手順

- 回路ファイルを解析し、
  各ゲート、信号線、ポート情報を構造体に格納
  - flex(字句解析)、bison(構文解析)

- ポート同士の接続関係構築(双方向接続)
  - buildPortConnections(module *m)
  - 各ポートの ”signal” 名をキーとして接続関係構築
- 外部入出力とポートの接続関係構築(双方向接続)
  - connectExternalPorts(module *m)
  - 各ポートの ”signal” 名をキーとして接続関係構築

接続のための関数

- 例)
  - NAND2_X1　G1　(.A1(A), .A2(B), .ZN(w1));
  - XNOR2_X1　G2　(.A(B), .B(C), .ZN(w2));
    - signal “B” をキーとし、”G1->.A2” と “G2->.A” を双方向接続

各構造体とその要素の接続関係
※port を介して 全netlist 走査可能