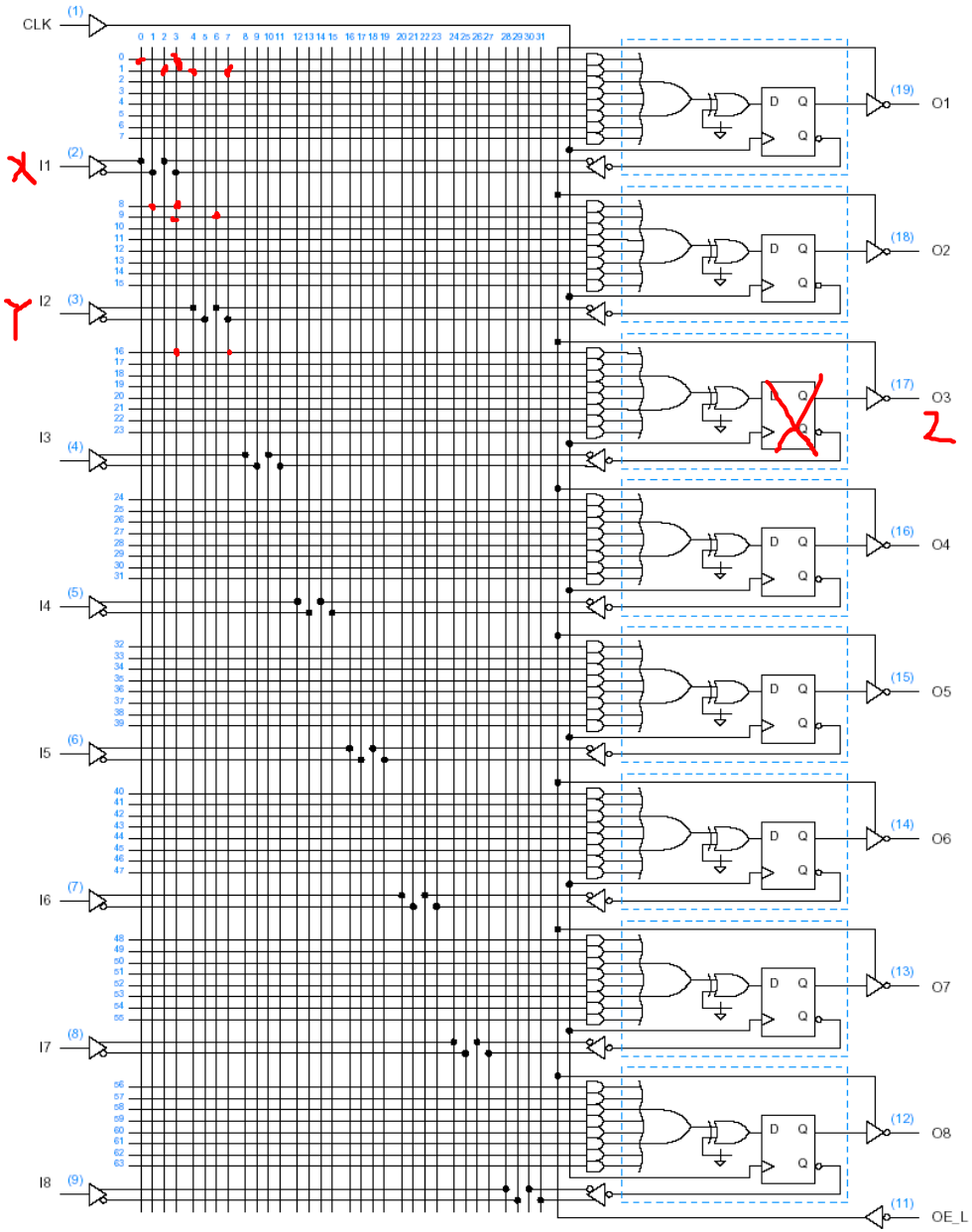
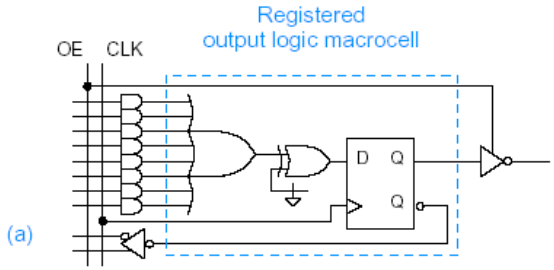
Chapter 10. CPLD, FPGA

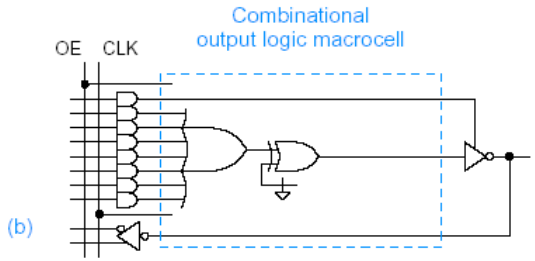
Modern Design Method

- **Using CAD tools (like Xilinx) and programmable logic devices**
  - Along with RAMs and ROMs
  - Complex Programmable Logic Devices (CPLDs)
  - Field Programmable Logic Devices (FPGAs)
- **Xilinx XC9500 CPLD and XC4000 FPGA family**
  - Can program wide range of combinational circuits
  - Can program wide range of sequential circuits
  - CAD tool automatically do (fitting/place-and-route) from our high level specification

PLD (Programmable Logic Device)





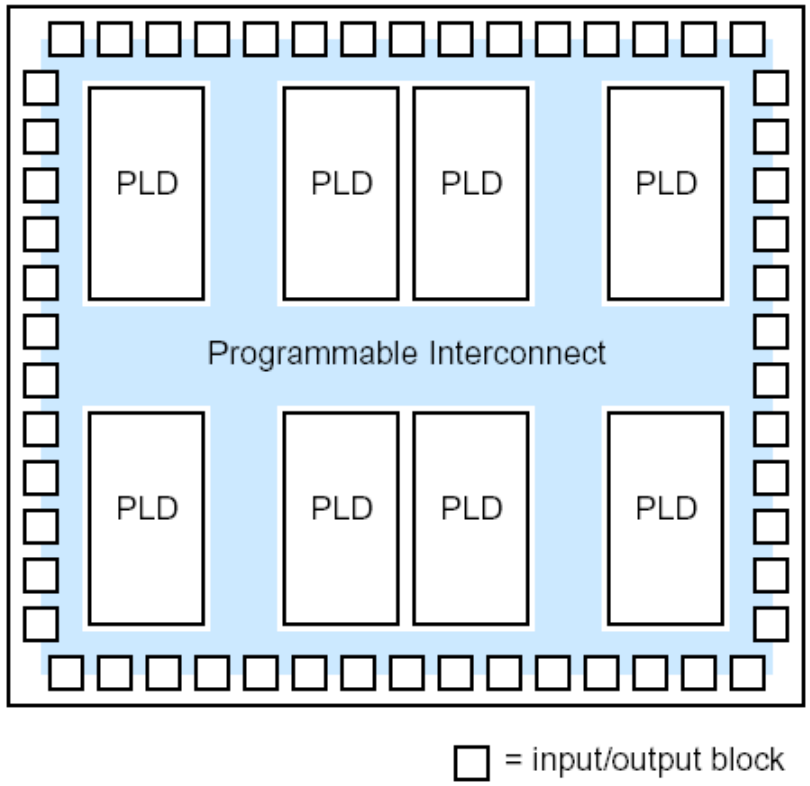


$$D_1 = Q_1 X + Q_2 \overline{Q}_1 Y$$

$$D_2 = Q_1 \overline{X} + \overline{Q}_2 Q_1$$

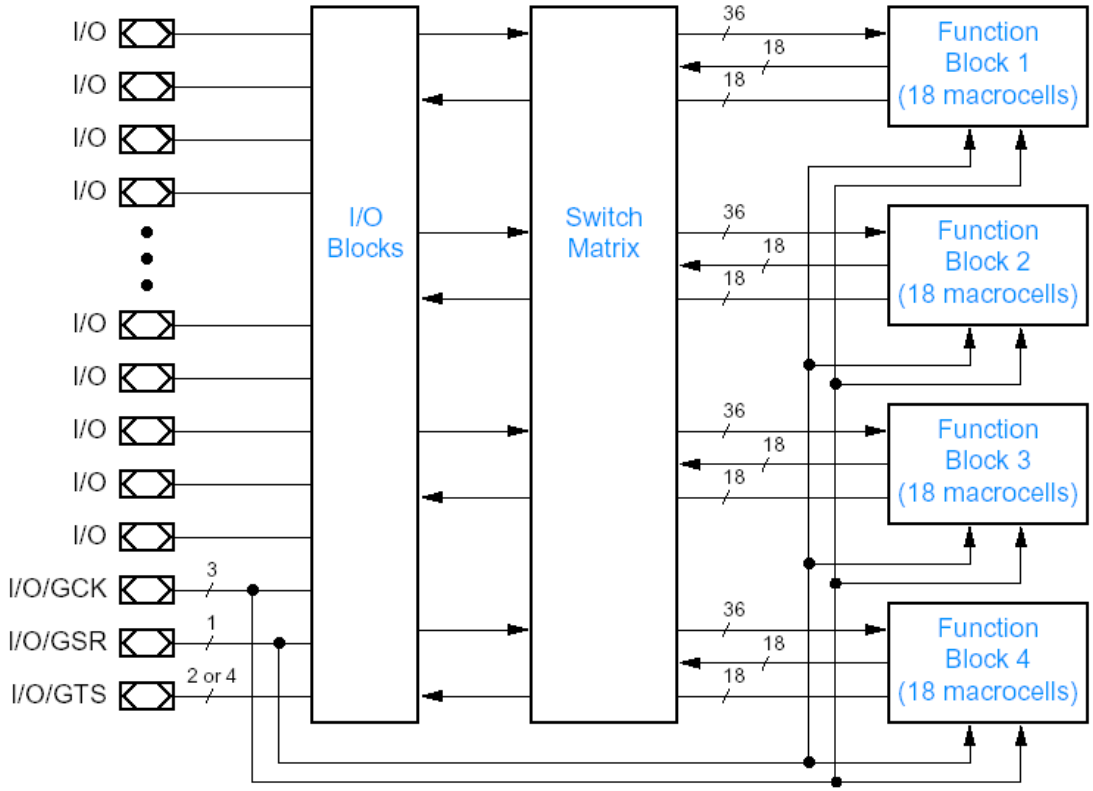
$$Z = Q_1 Q_2$$

General CPLD Architecture

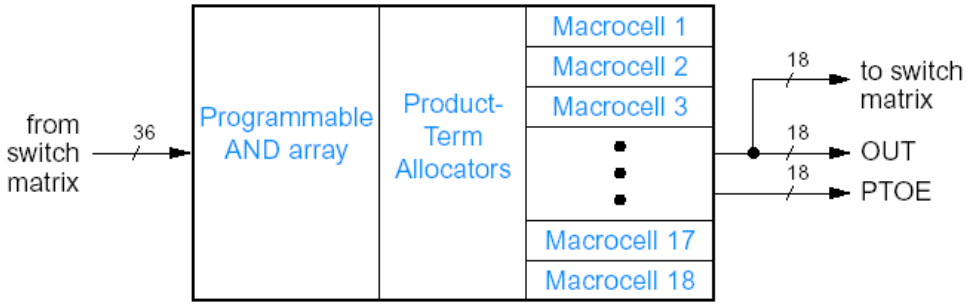


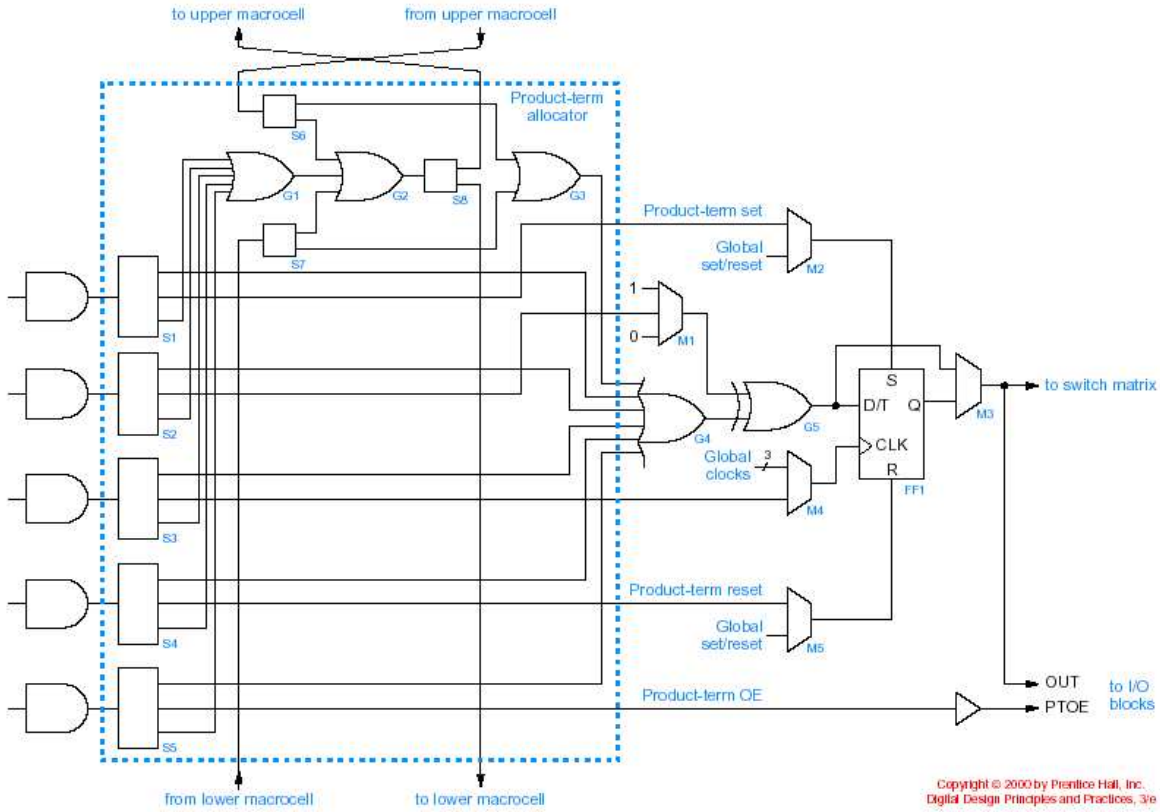
- **PLD**
- **Programmable Interconnect**
- **I/O block**

Xilinx 9500-family architecture

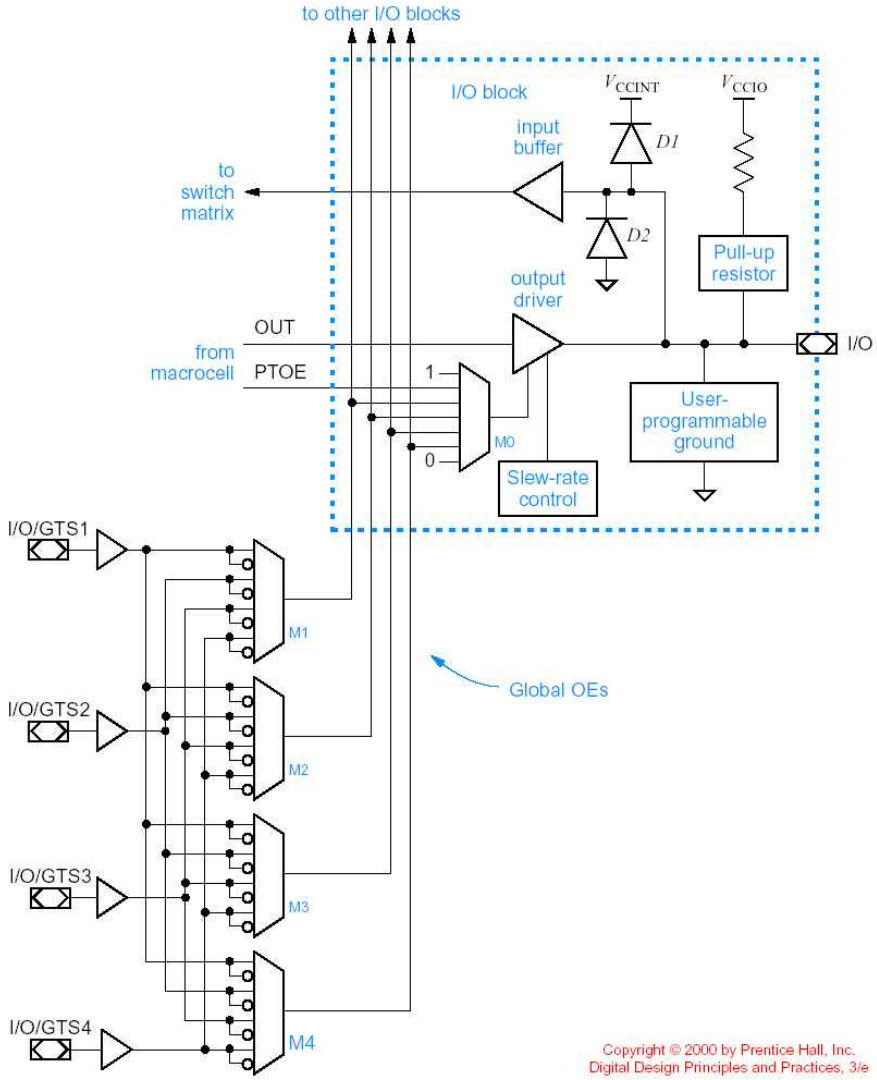


Function Block

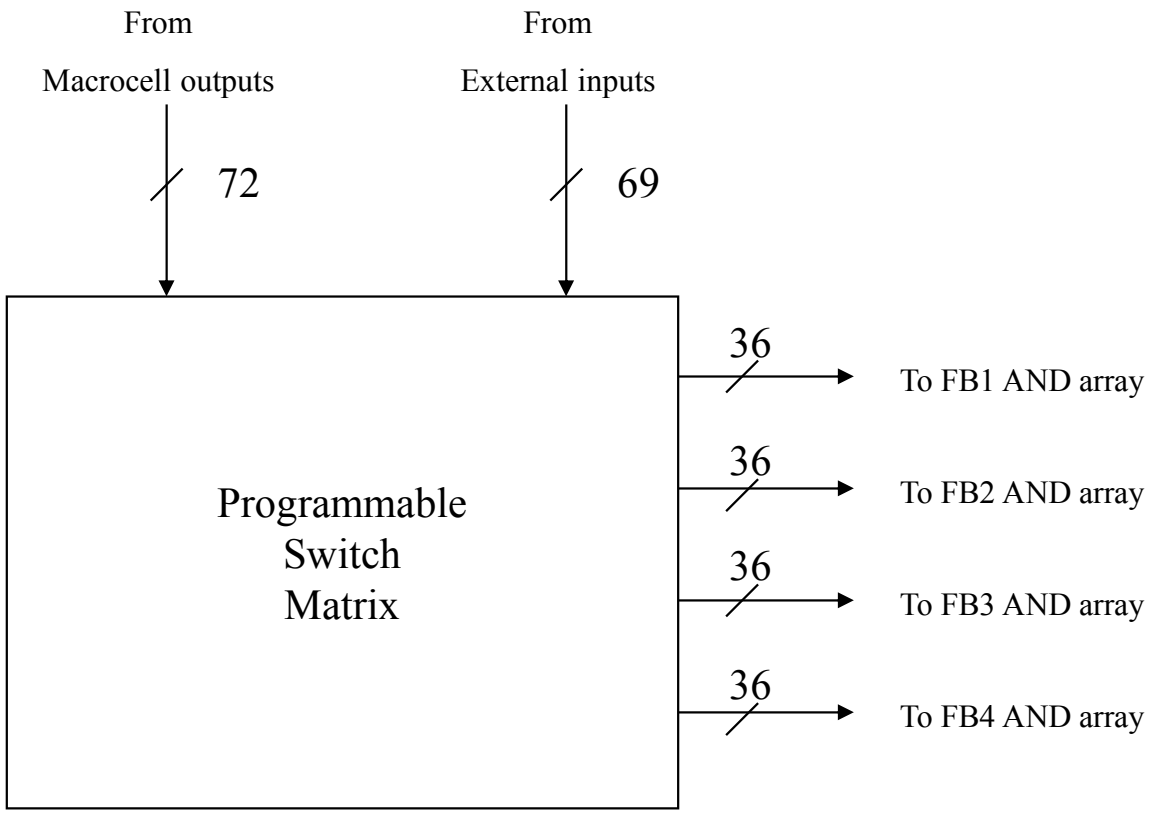




I/O Block



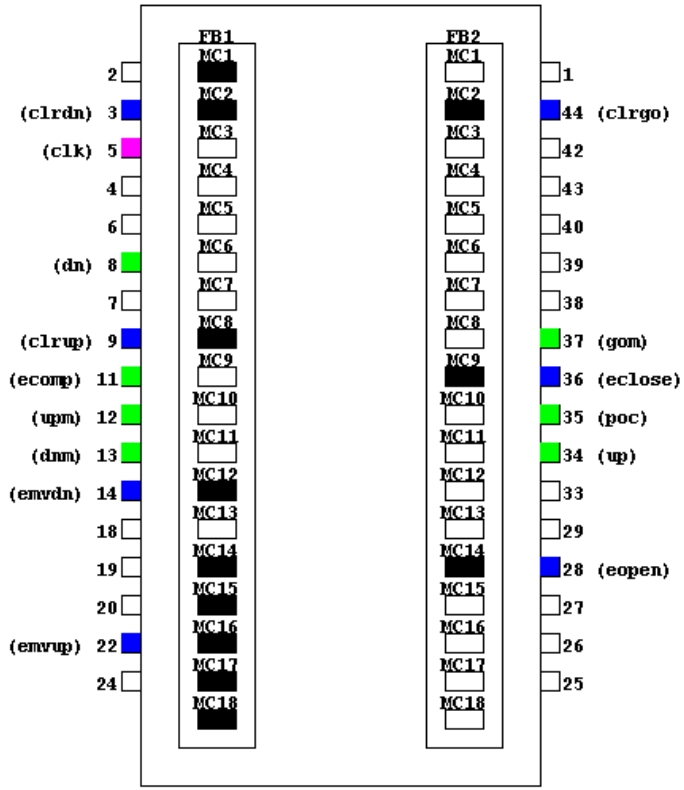
Switch Matrix



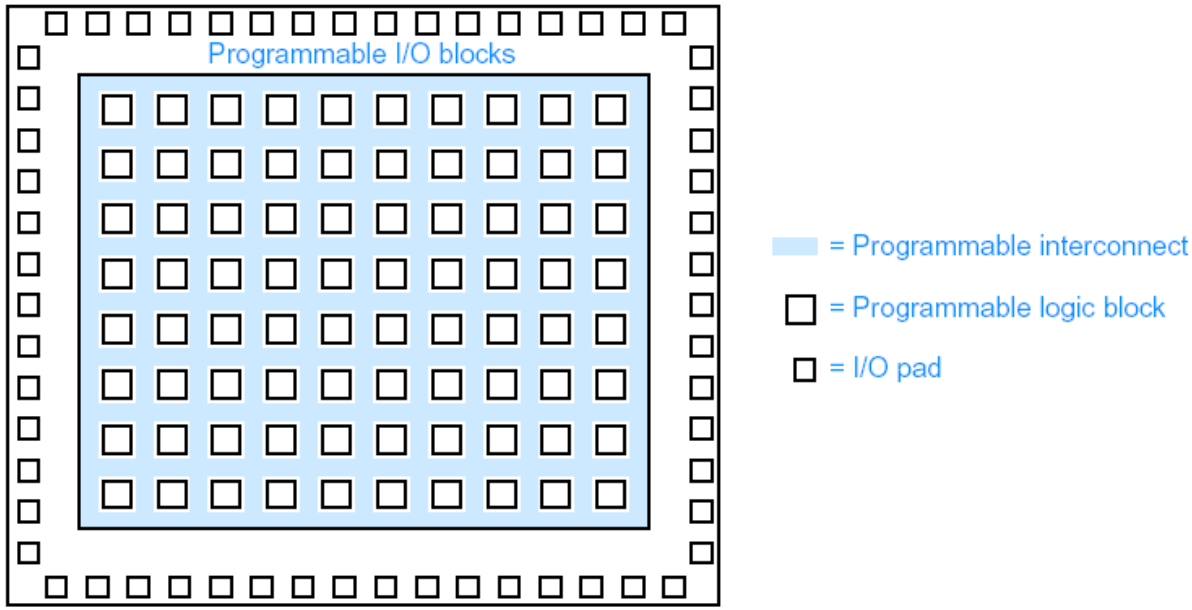
Programmable elements

- **Many elements should be correctly programmed**
  - FBs
    - Programmable AND array
    - Product term allocator
    - Programmable Mux in macrocells
  - I/O Blocks
    - Output enable selection for three-state driver output buffer
  - Switch matrix
    - Programmable connection between switch inputs (macrocell outputs and external inputs) to switch outputs (inputs to FB AND array)
- **Fortunately, once we give a high-level design description (using schematic form and VHDL, etc), CAD's fitting software automatically find the solution**

View of Fitted Design
- XC9536-5-PC44 -
(elevator system controller)

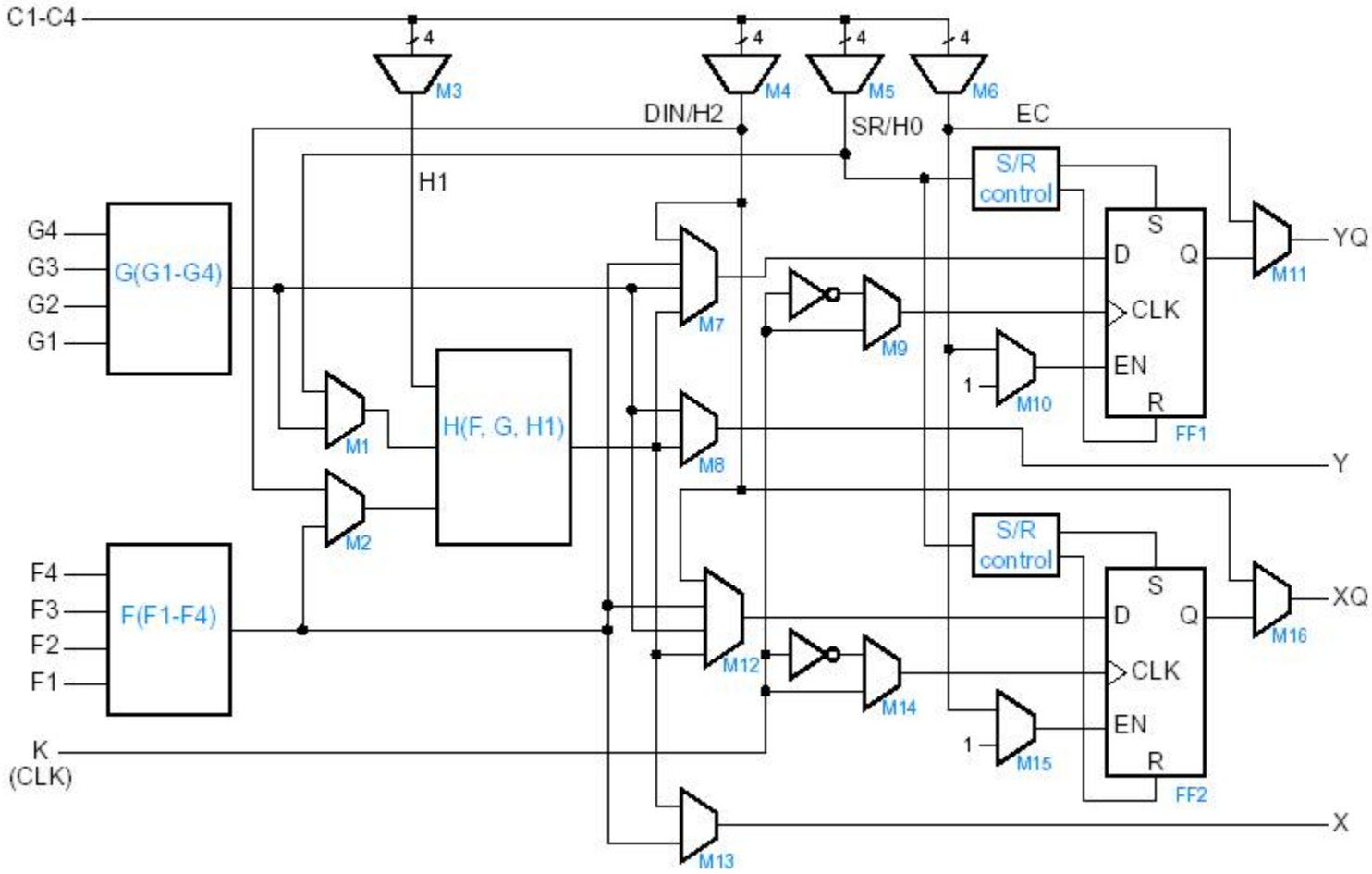


General FPGA chip architecture



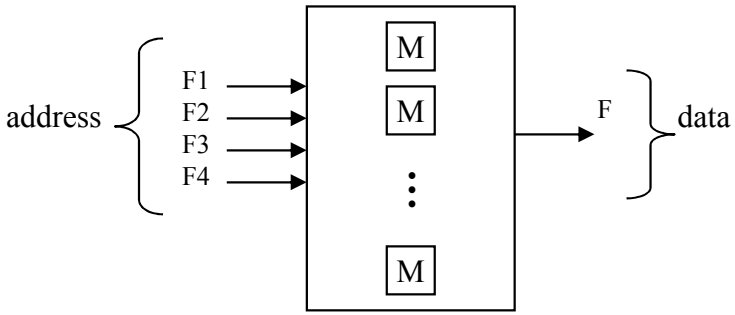
- **Larger number of small blocks (compare with CPLD)**
- **3 basic components**
  - Configurable Logic Block (CLB) – programmable logic
  - Input/Output Block (IOB) – around chip, associated with I/O pins of the package
  - Programmable interconnects – interconnect CLBs and IOBs for implementing larger functions

Configurable Logic Block (CLB)



3 Look Up Tables (G, F, H)

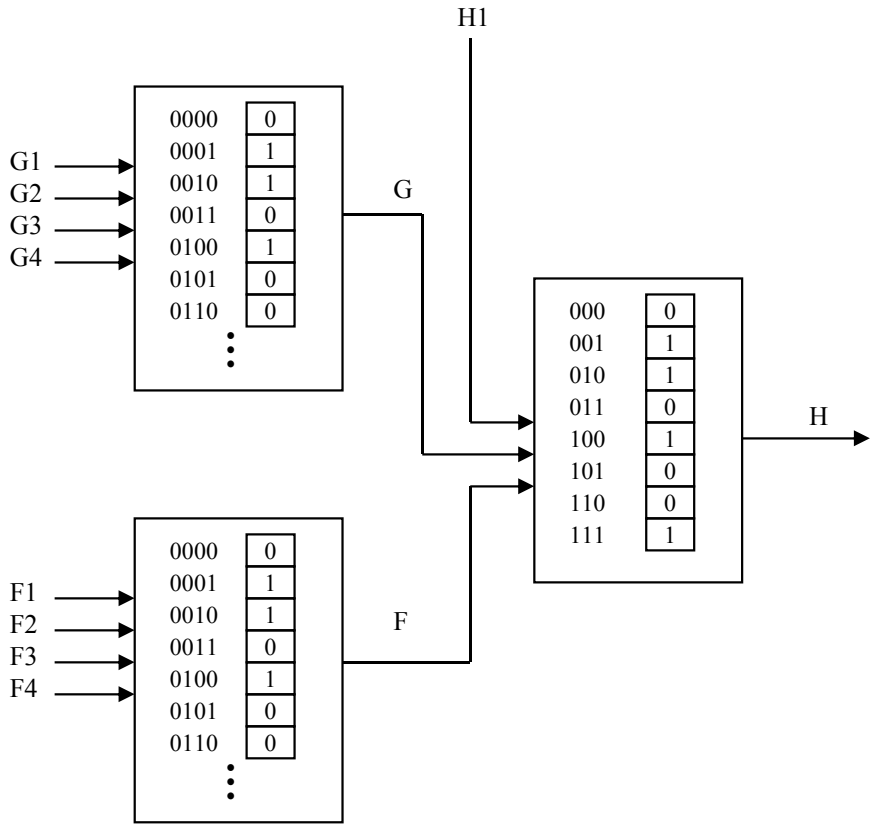
- **Configured to implement any function of 4 (3 for H) inputs**
- **Universal Function Generator**
  - How to build?
  - With SRAM



- 16x1 SRAM
- Store truth table of the Function (2^4 rows)
- SRAM loaded at config time – to make it work as a specific function

Example Function using F, G, H

- **Parity of 9 inputs: H = 1 if odd, H=0 if even**



Power of F, G, H

$$G = G(G1, G2, G3, G4)$$
$$F = F(F1, F2, F3, F4)$$
$$H = H((F, H2), (G, H0), H1)$$

- ***Any*** **function of 4 inputs (F) +** ***Any*** **function of 4 inputs (G) +** ***Any*** **function of 3 inputs (H)**
- ***Any*** **function of 5 inputs (F+G+H) – how?**
- ***Any*** **function of 4 inputs (F) +** ***Some*** **function of 6 inputs**
- ***Some*** **function of 9 inputs (F+G+H)**

2 D f/fs

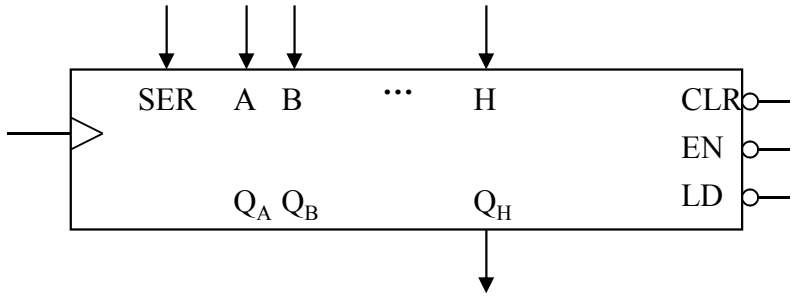
- **D: F or G or H or DIN/H2 (Dx, Dy configured by M12, M7)**
- **CLK: K or K- (CLK_x, CLK_y configured by M14, M9)**
- **EN: 1 or EC (ENx, ENy configured by M15, M10)**
- **S/R control**
  - set or reset at configuration
  - Respond to global set/reset signal (not shown) or S/R line

Outputs

- **X = F or H**
- **XQ = M4 or Qx (Dx = F or G or H or DIN)**
- **Y = G or H**
- **YQ = M6 or Qy (Dy = F or G or H or DIN)**

Example Configuration of CLBs

- **Implementing 74x166**
  (8-bit parallel-in, serial-out shift register)



$$D_A = \overline{EN} \cdot Q_A + EN \cdot \overline{LD} \cdot SER + EN \cdot LD \cdot A$$

$$D_B = \overline{EN} \cdot Q_B + EN \cdot \overline{LD} \cdot Q_A + EN \cdot LD \cdot B$$

$$\vdots$$

$$D_H = \overline{EN} \cdot Q_H + EN \cdot \overline{LD} \cdot Q_G + EN \cdot LD \cdot H$$

Can implement any 5 variable functions?

$$D_A = \overline{EN} \cdot Q_A + EN \cdot \overline{LD} \cdot SER + EN \cdot LD \cdot A$$

$$H1 = Q_A$$

$$G = D_A \text{ (when } H1 = 0) = EN \cdot \overline{LD} \cdot SER + EN \cdot LD \cdot A$$

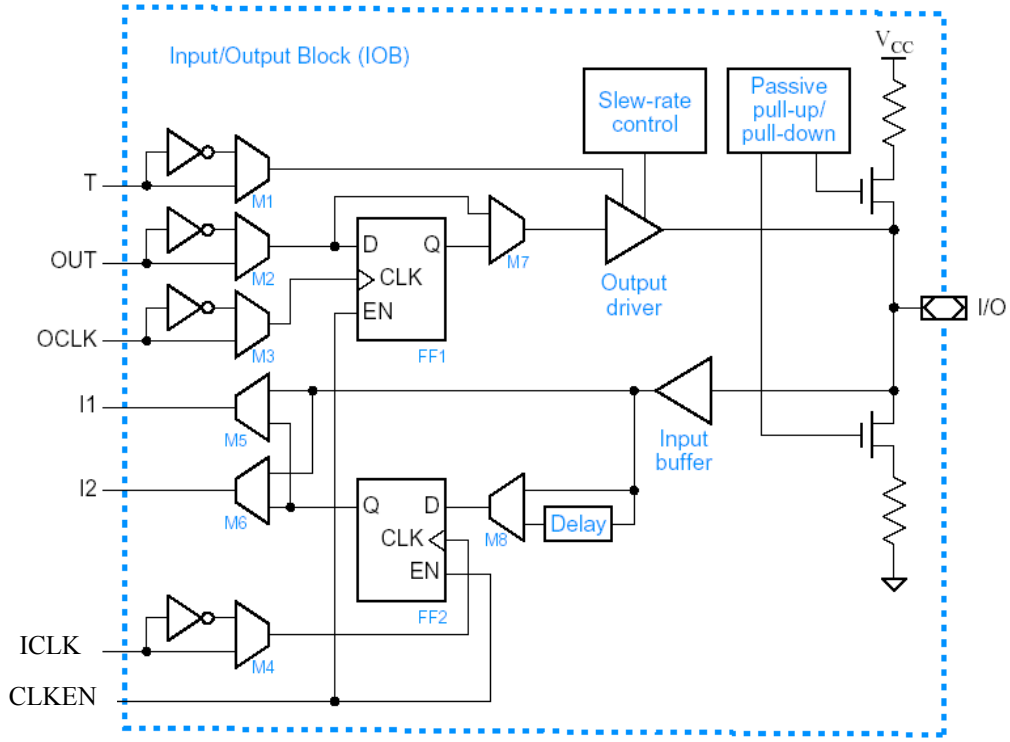
$$F = D_A \text{ (when } H1 = 1) = \overline{EN} + EN \cdot \overline{LD} \cdot SER + EN \cdot LD \cdot A$$

$$H = \overline{H1} \cdot G + H1 \cdot F$$

How many CLBs for implementing 8-bit shift register x166?

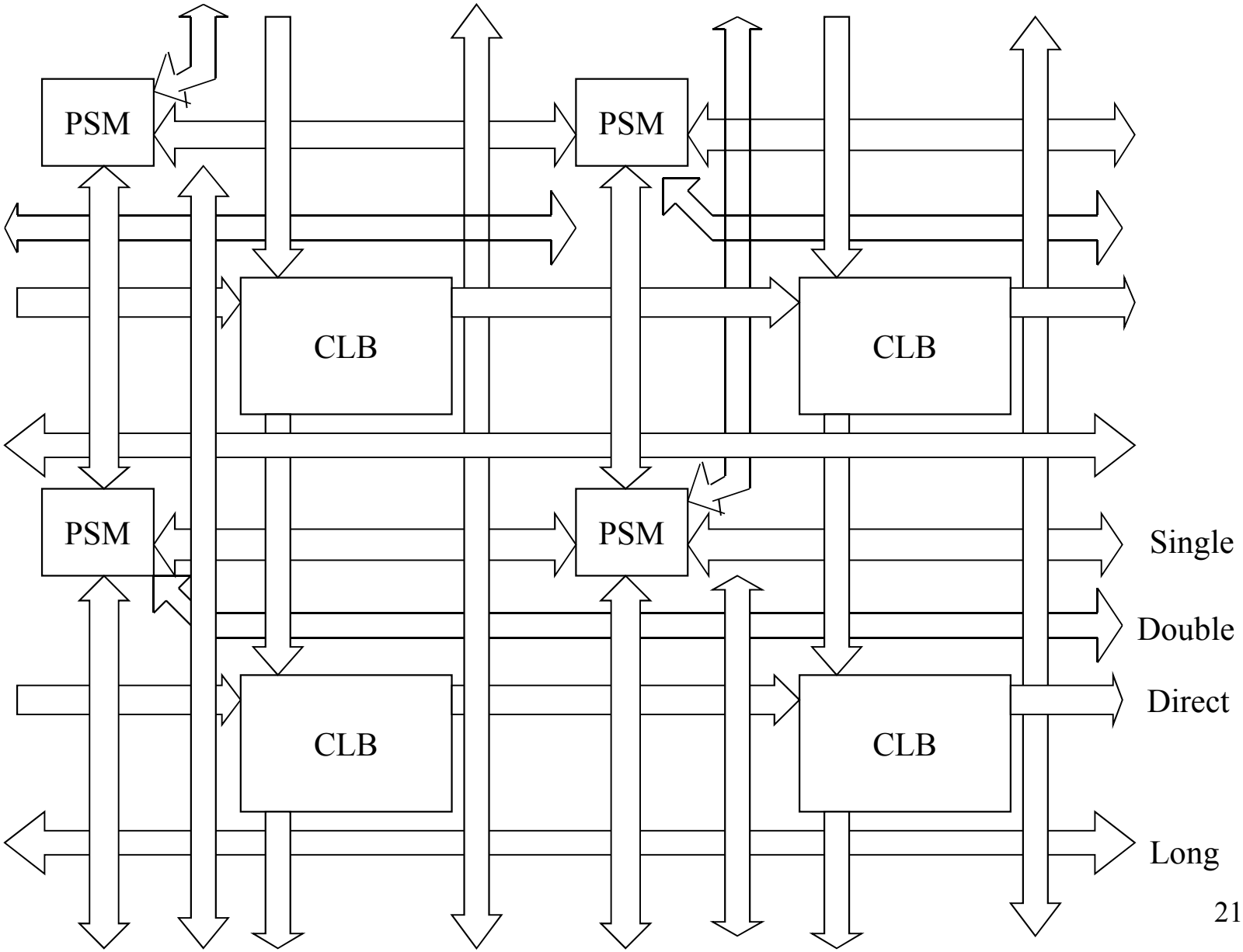
- only 8 f/fs $\rightarrow$ 4 CLBs?
- But, all G, F, H in a CLB should be used for each D equation.

I/O Block (IOB)

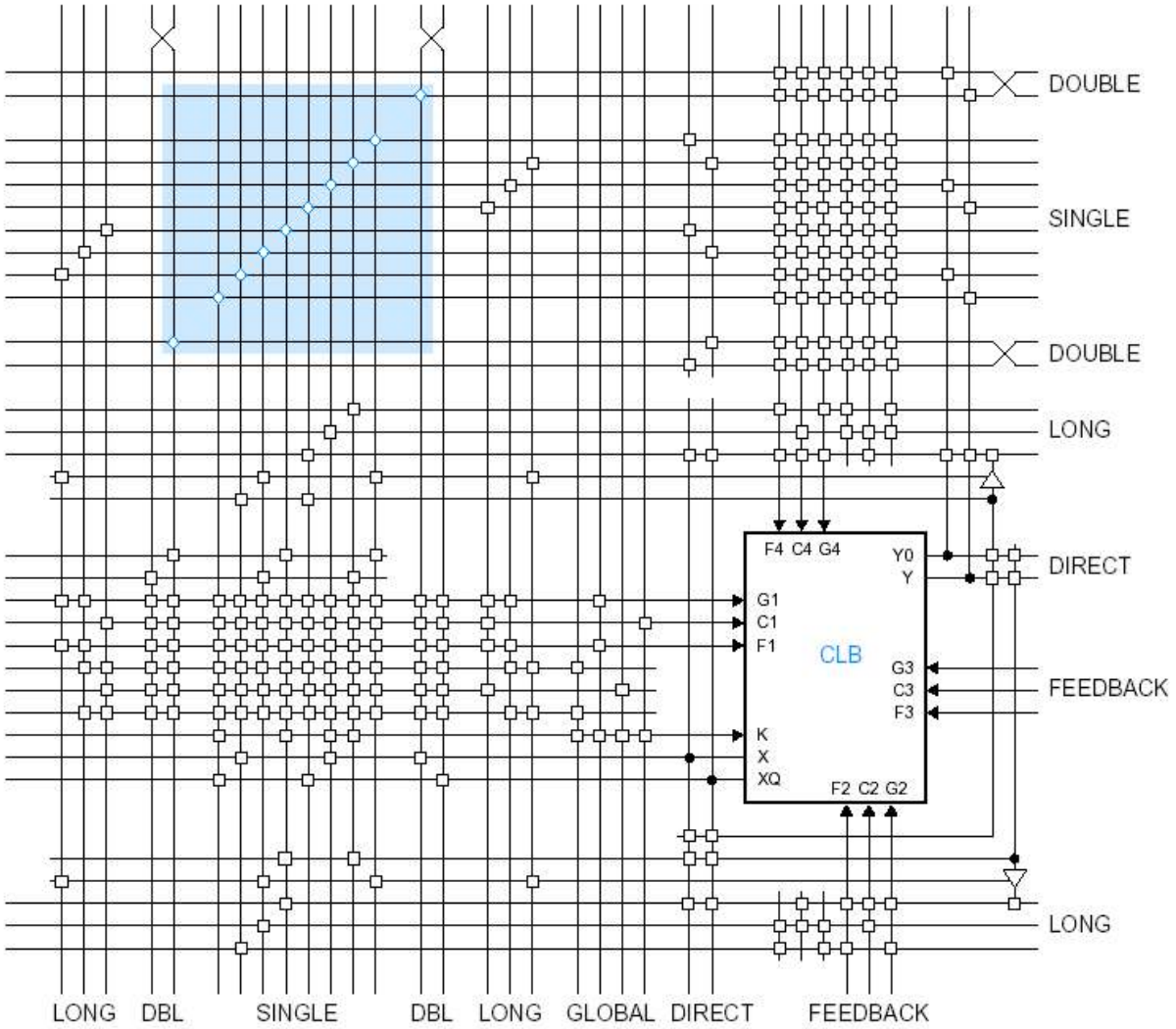


- **1 per I/O pin**
- **2 f/fs**
- **Open collector, tri-state output, registered or not**
- **Direct Input or Input through f/f (Synchronizer)**

Programmable Interconnect

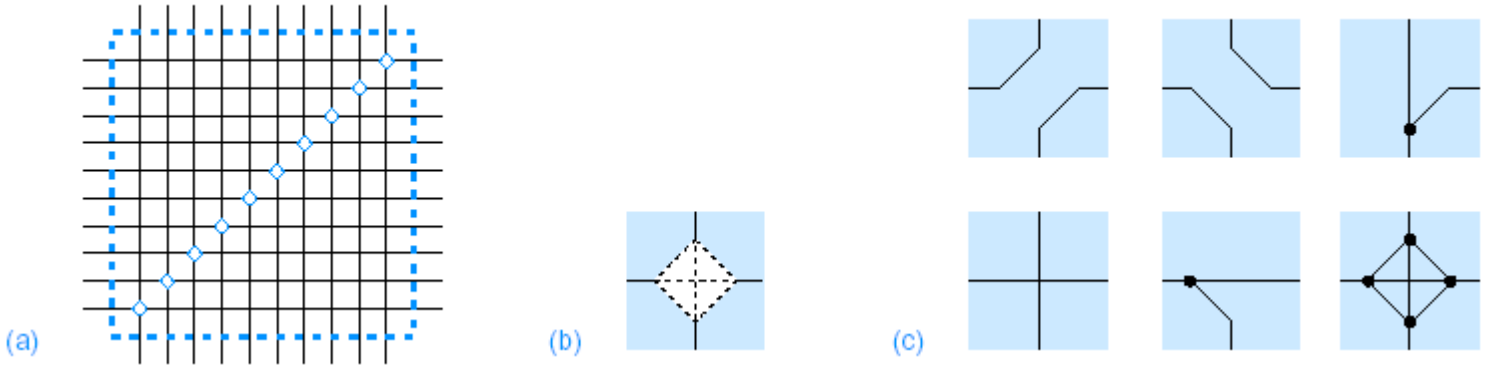


Detail Look


DOUBLE
SINGLE
DOUBLE
LONG
DIRECT
FEEDBACK
LONG
F4 C4 G4
Y0
Y
G1
C1
F1
CLB
G3
C3
F3
K
X
XQ
F2 C2 G2
LONG
DBL
SINGLE
DBL
LONG
GLOBAL
DIRECT
FEEDBACK

Programmable Switch Matrix (PSM)

- **PSM make possible variety of difficult interconnections**
  - Can lengthen segments
  - Can turn corners



Pretty Complex Interconnect

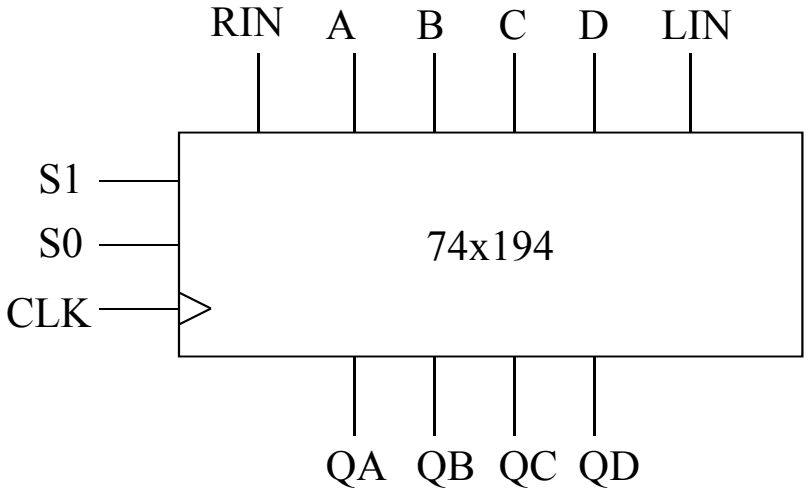
- **Implement huge combination of Interconnects using**
  - Direct wires – direct connect to adjacent CLB
  - Single wires – one hop connect to adjacent CLB through a PSM
  - Double wires – travel past 2 CLBs before hitting a switch
  - Long wires – travel length of chip
  - Programmable connections
  - Programmable switch matrix (PSM)

Place and Routing

- **Place (On which CLB, a specific function need to be implemented)**
- **Routing (How to interconnect CLBs?)**
  - Avoid use of PSM as much as possible to reduce the delay
- **Place and Routing is an inter-dependent complex problem**
- **Fortunately, CAD tool will do for us**

Example

- **Find Place and Routing for 74x194**



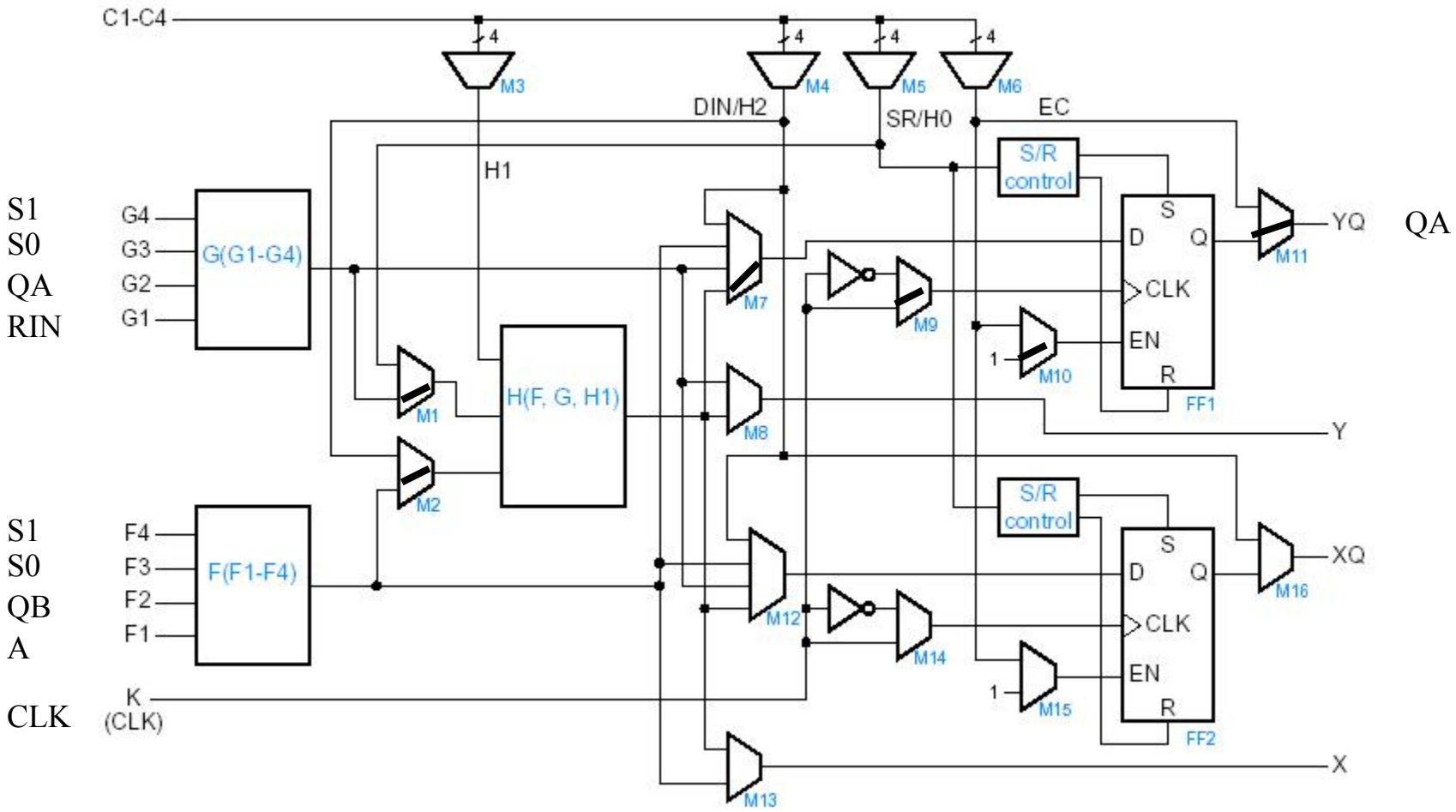
$$DA = \overline{S1}\,\overline{S0}QA + \overline{S1}S0RIN + S1\overline{S0}QB + S1S0A$$

$$DB = \overline{S1}\,\overline{S0}QB + \overline{S1}S0QA + S1\overline{S0}QC + S1S0B$$

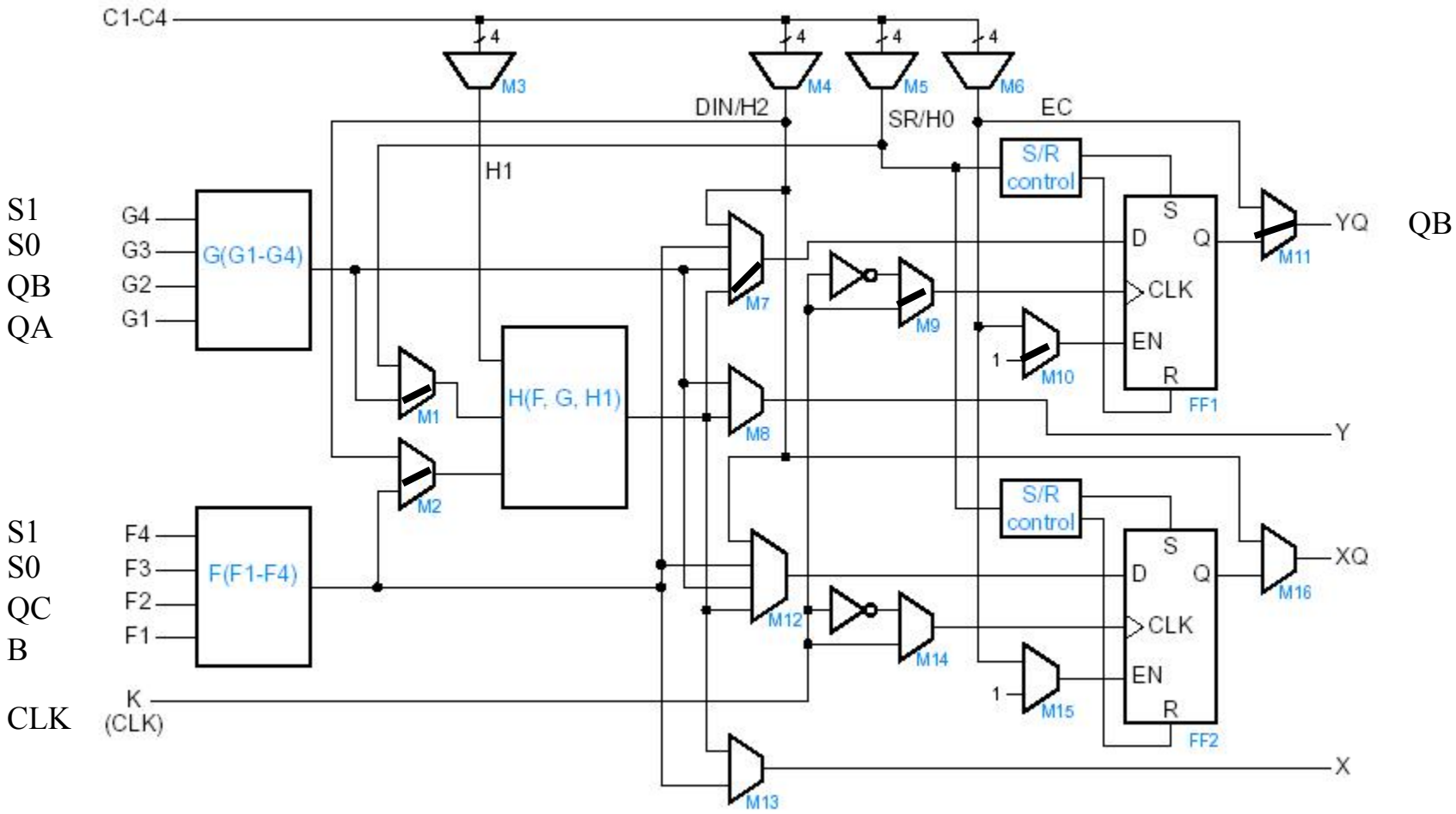
$$DC = \overline{S1}\,\overline{S0}QC + \overline{S1}S0QB + S1\overline{S0}QD + S1S0C$$

$$DD = \overline{S1}\,\overline{S0}QD + \overline{S1}S0QC + S1\overline{S0}LIN + S1S0D$$

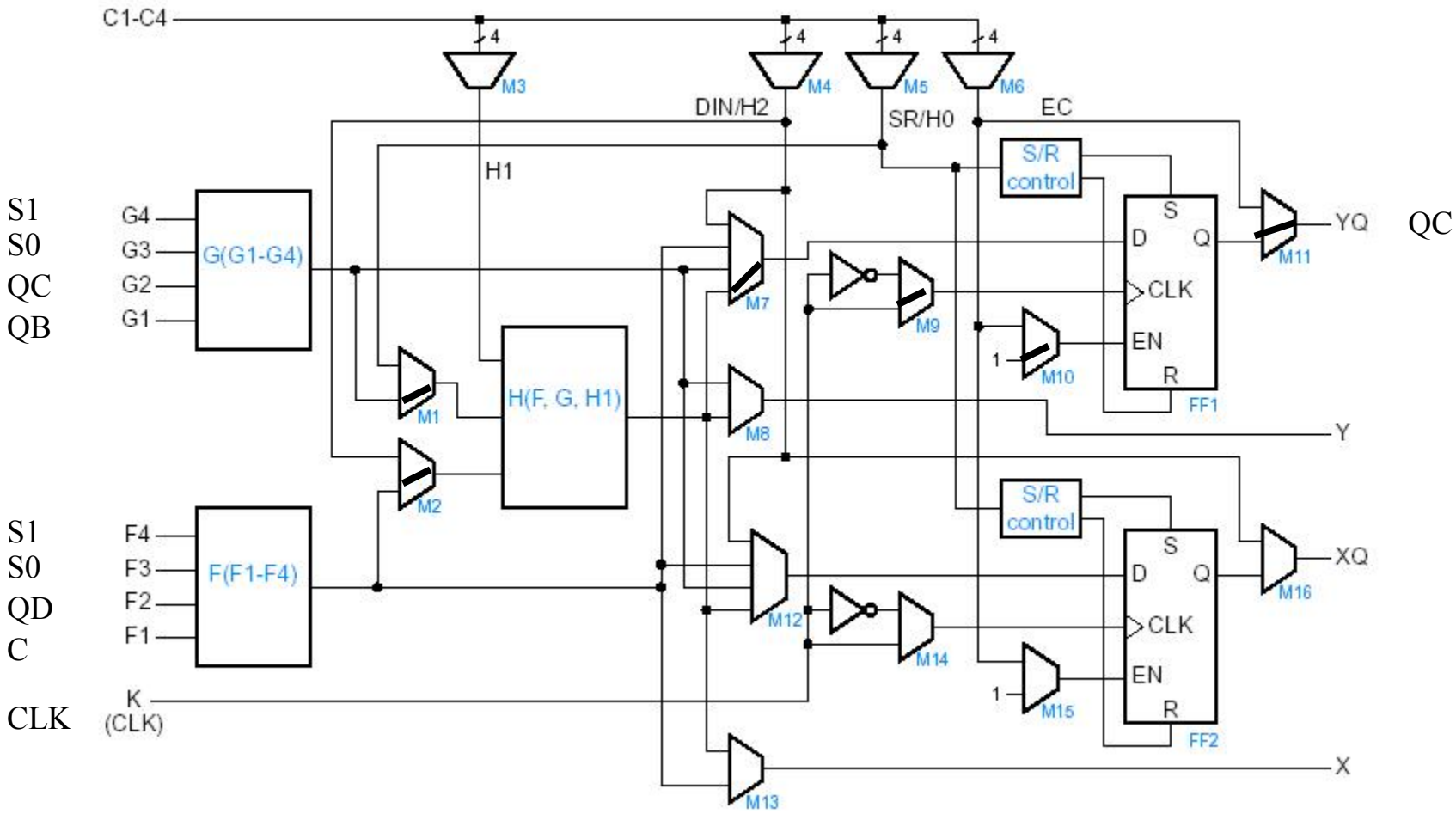
QA


C1-C4
4
M3
M4
M5
M6
DIN/H2
SR/H0
EC
H1
S1
S0
QA
RIN
G4
G3
G2
G1
G(G1-G4)
M1
M2
H(F, G, H1)
M7
M8
M9
M10
S/R control
S
D
Q
CLK
EN
R
FF1
M11
YQ
QA
Y
S1
S0
QB
A
F4
F3
F2
F1
F(F1-F4)
M12
M13
M14
M15
M16
FF2
XQ
X
CLK
K
(CLK)
1

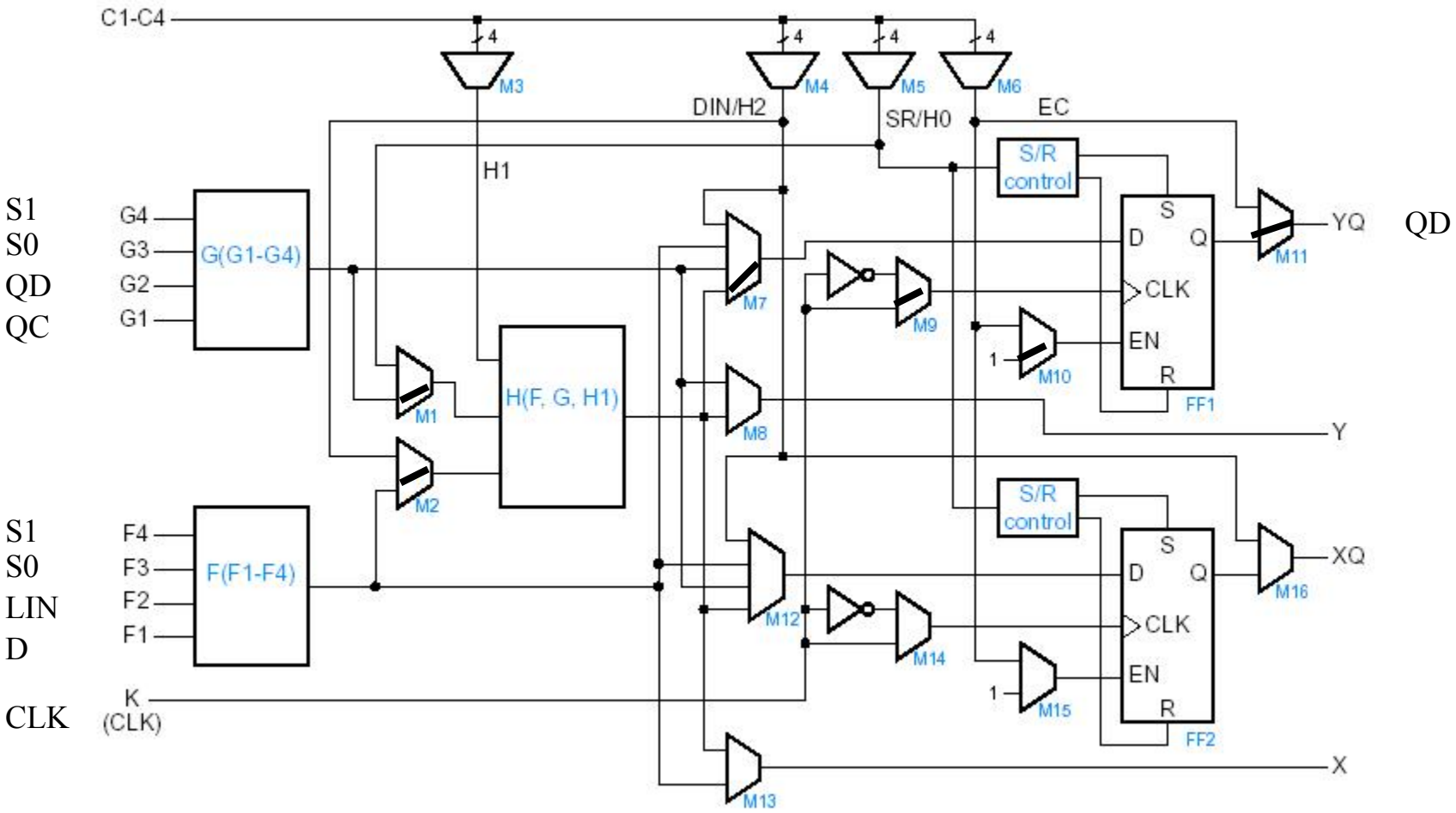
QB

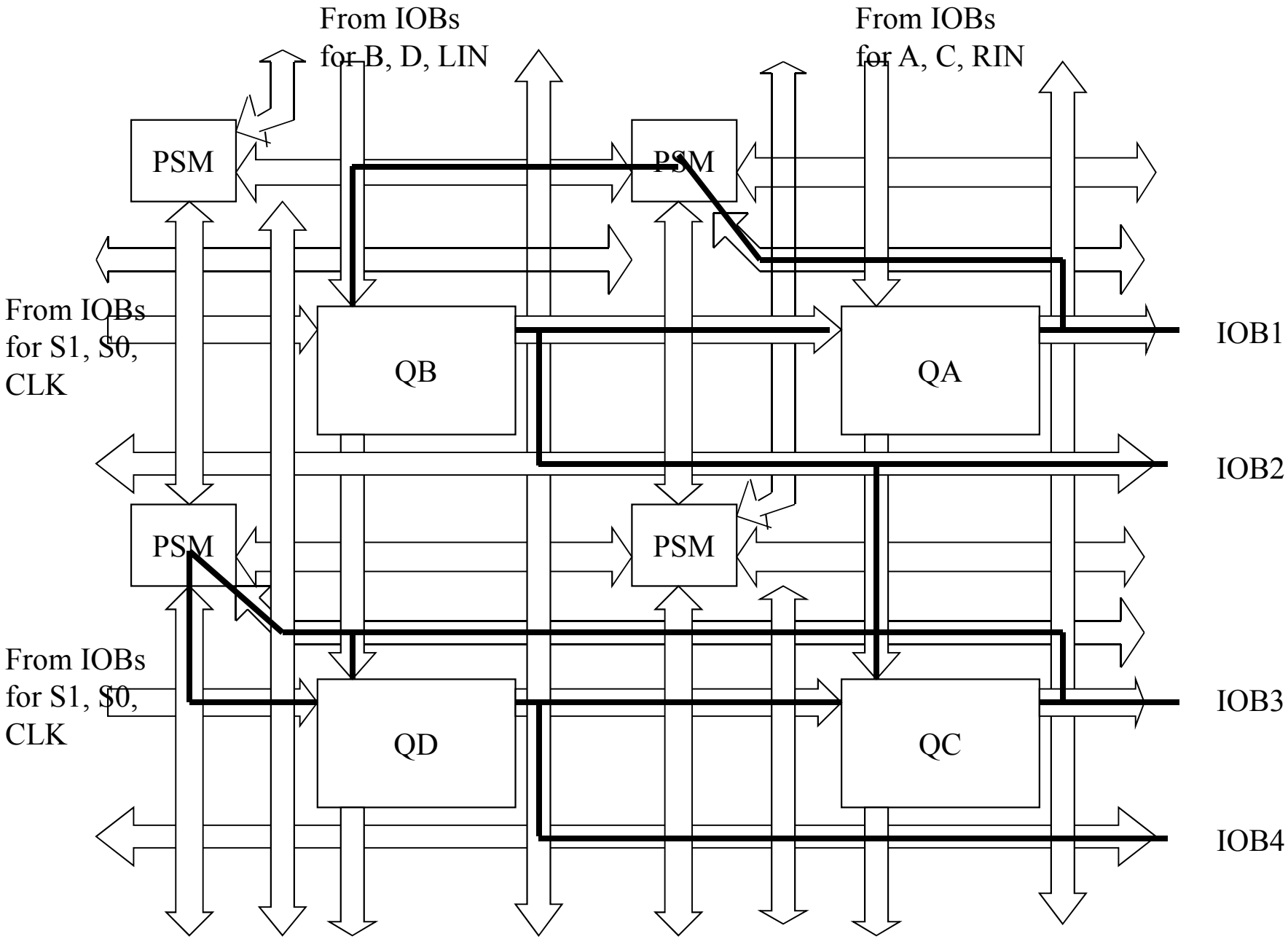

C1-C4
4
M3
M4
M5
M6
DIN/H2
SR/H0
EC
H1
S1
S0
QB
QA
G4
G3
G2
G1
G(G1-G4)
M1
M2
H(F, G, H1)
M7
M8
M9
M10
S/R control
S
D
Q
CLK
EN
R
FF1
M11
YQ
QB
Y
S1
S0
QC
B
F4
F3
F2
F1
F(F1-F4)
M12
M13
M14
M15
M16
XQ
X
CLK
K
(CLK)
1
FF2

QC


C1-C4
4
M3
M4
M5
M6
DIN/H2
SR/H0
EC
H1
S1
S0
QC
QB
G4
G3
G2
G1
G(G1-G4)
M1
M2
H(F, G, H1)
M7
M8
M9
M10
S/R control
S
D
Q
CLK
EN
R
FF1
M11
YQ
QC
Y
S1
S0
QD
C
F4
F3
F2
F1
F(F1-F4)
M12
M13
M14
M15
M16
FF2
XQ
X
CLK
K
(CLK)
1

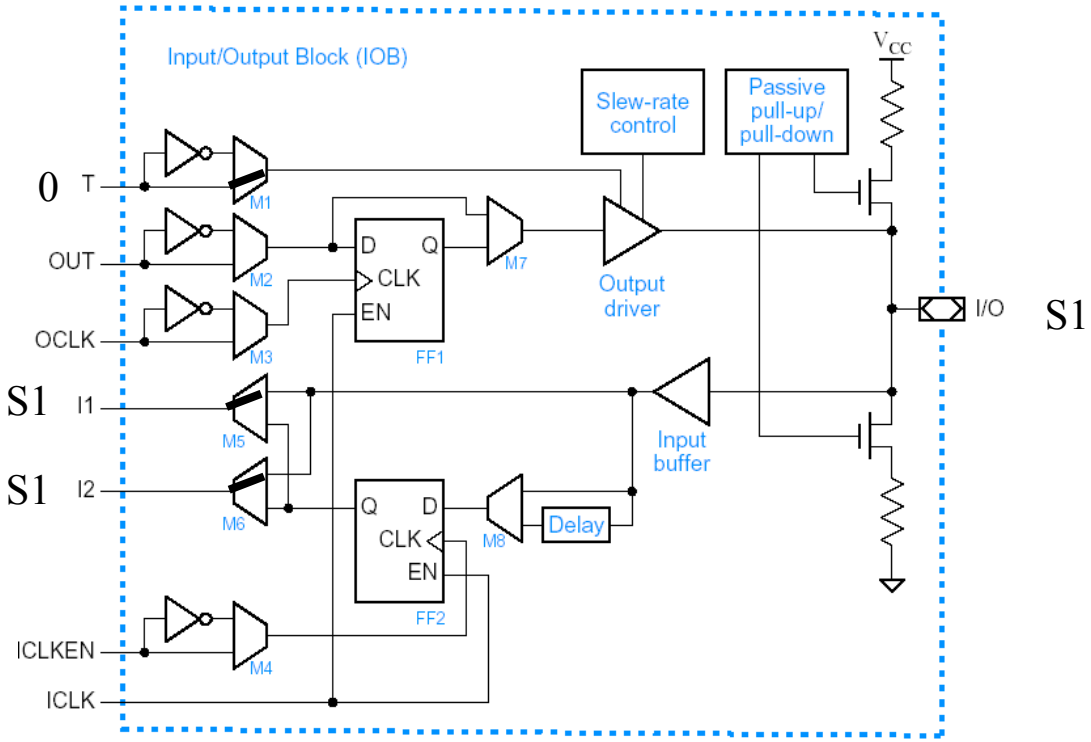
QD


C1-C4
4
M3
M4
M5
M6
DIN/H2
SR/H0
EC
H1
S1
S0
QD
QC
G4
G3
G2
G1
G(G1-G4)
M1
M2
H(F, G, H1)
M7
M8
M9
M10
S/R control
S
D
Q
CLK
EN
R
FF1
M11
YQ
QD
Y
S1
S0
LIN
D
F4
F3
F2
F1
F(F1-F4)
M12
M13
M14
M15
M16
XQ
X
K
(CLK)
CLK
1

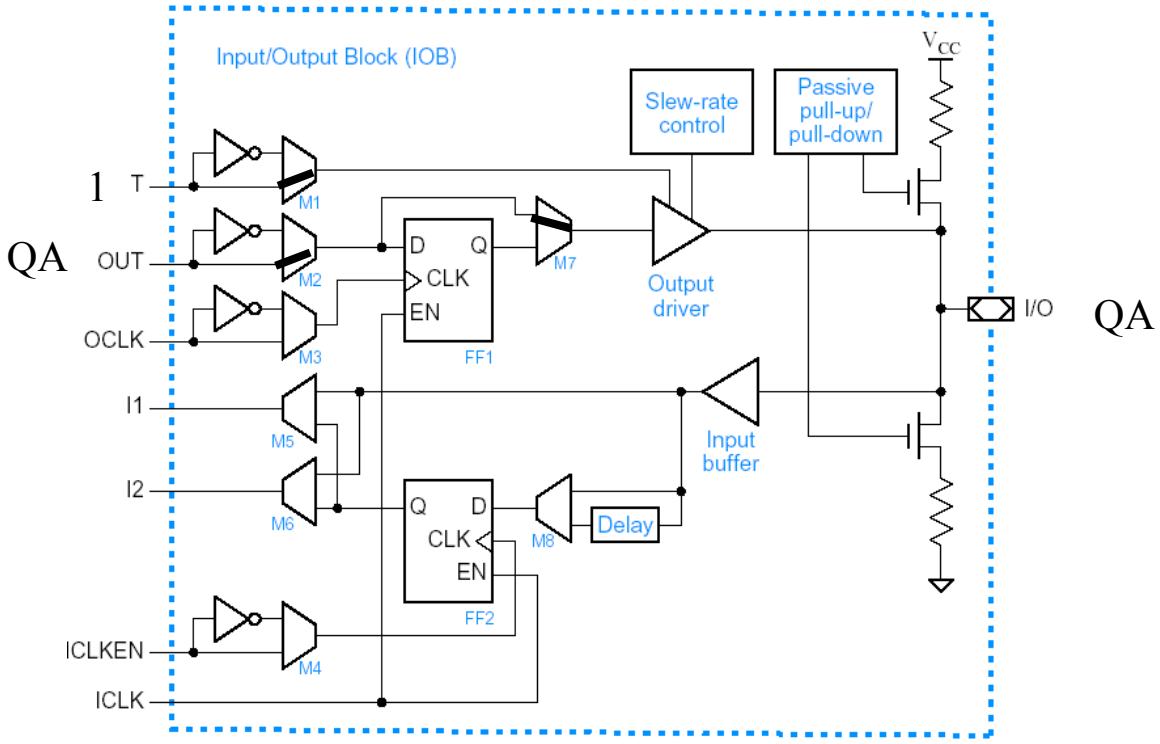


Routing from inputs is not shown

IOB for S1


Input/Output Block (IOB)
Slew-rate control
Passive pull-up/ pull-down
V_{CC}
0
T
M1
OUT
M2
OCLK
M3
D
Q
CLK
EN
FF1
M7
Output driver
I/O
S1
S1
I1
M5
S1
I2
M6
Input buffer
Q
D
CLK
EN
FF2
M8
Delay
ICLKEN
M4
ICLK

IOB for QA


Input/Output Block (IOB)
Slew-rate control
Passive pull-up/ pull-down
V_{CC}
1
T
M1
QA
OUT
M2
D
Q
CLK
EN
FF1
M7
Output driver
I/O
QA
OCLK
M3
I1
M5
Input buffer
I2
M6
Q
D
CLK
EN
FF2
M8
Delay
ICLKEN
M4
ICLK

View of Placed Design
- XC2VP2-7-FG256 –
(elevator system controller)