



Metodologia Projektowania na podstawie:

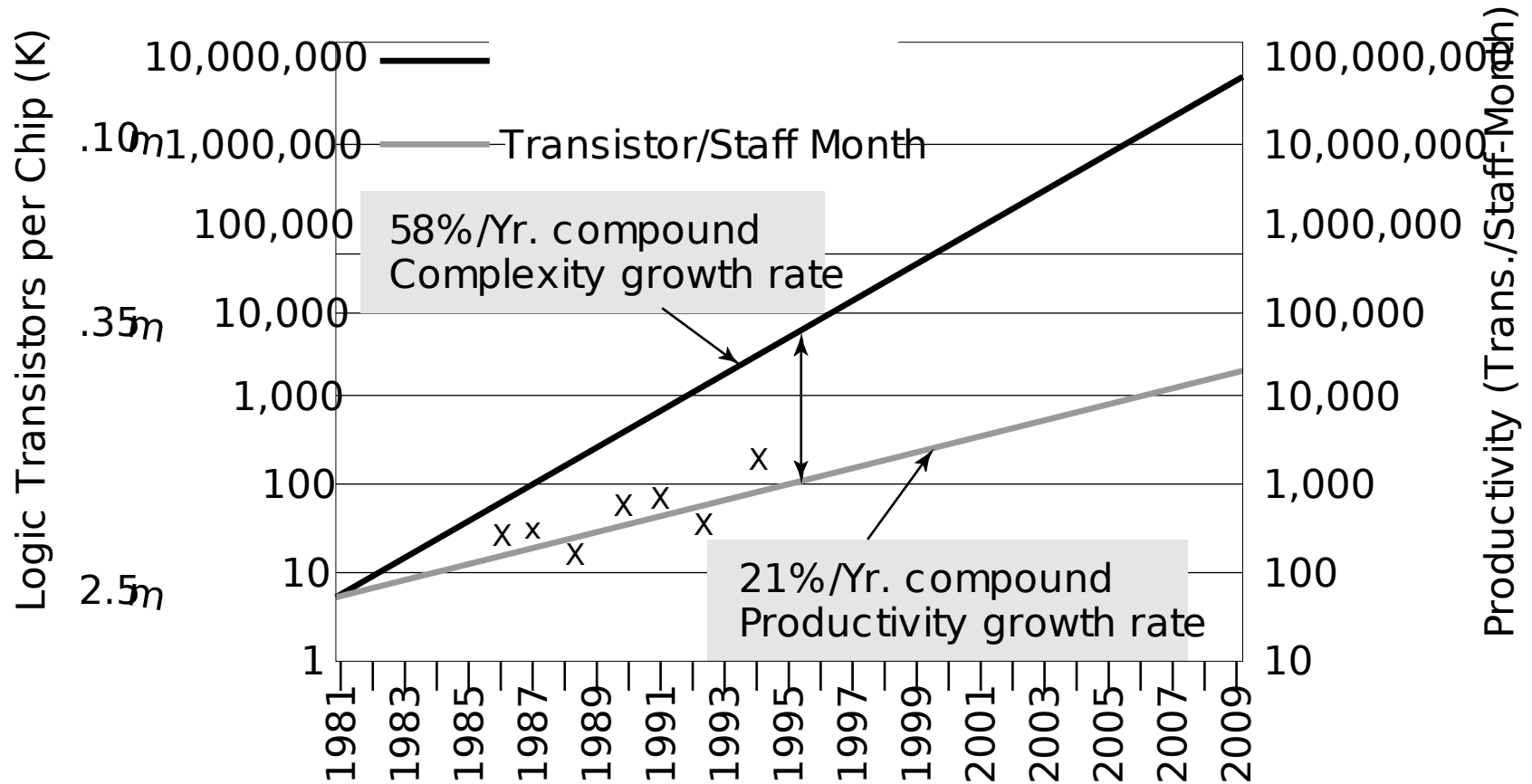
Digital Integrated Circuits

Jan M. Rabaey

Anantha Chandrakasan

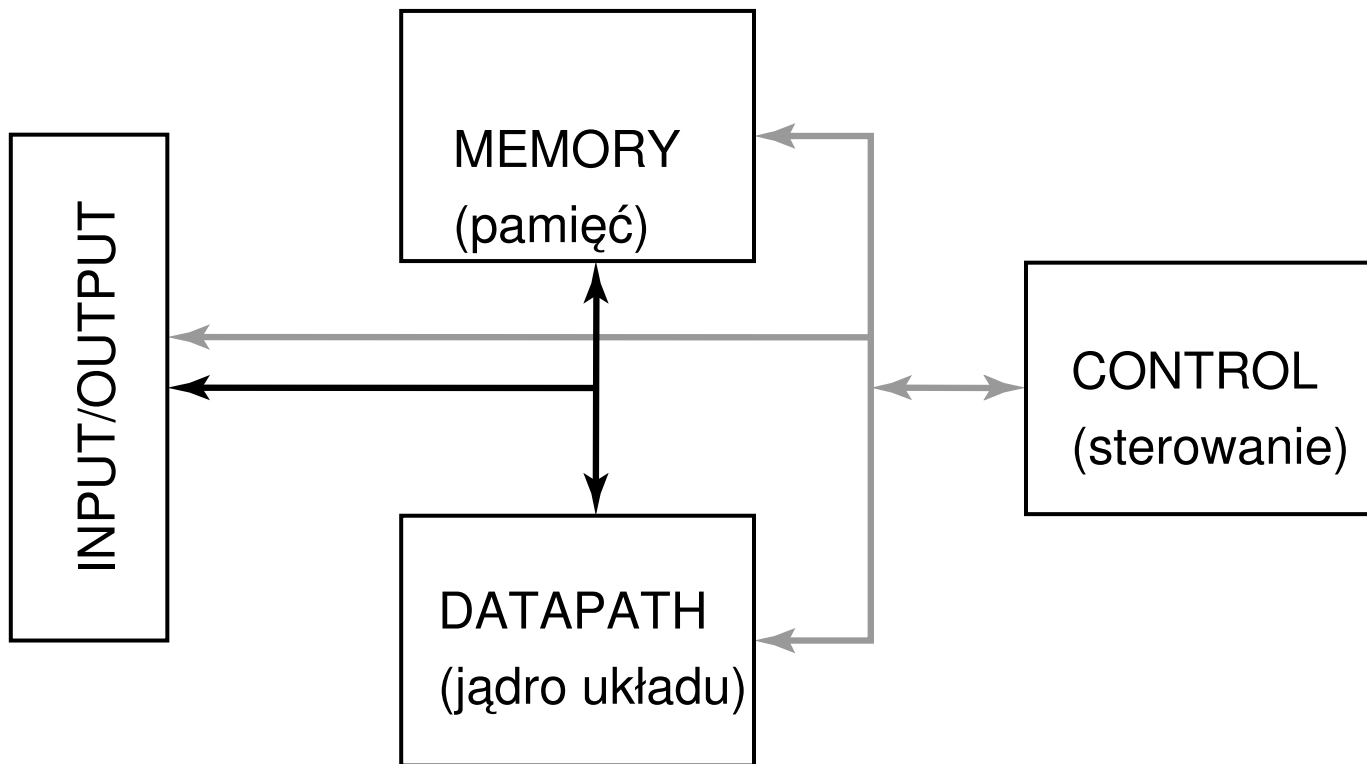
Borivoje Nikolic

Wyzwanie w projektowaniu elektroniki cyfrowej

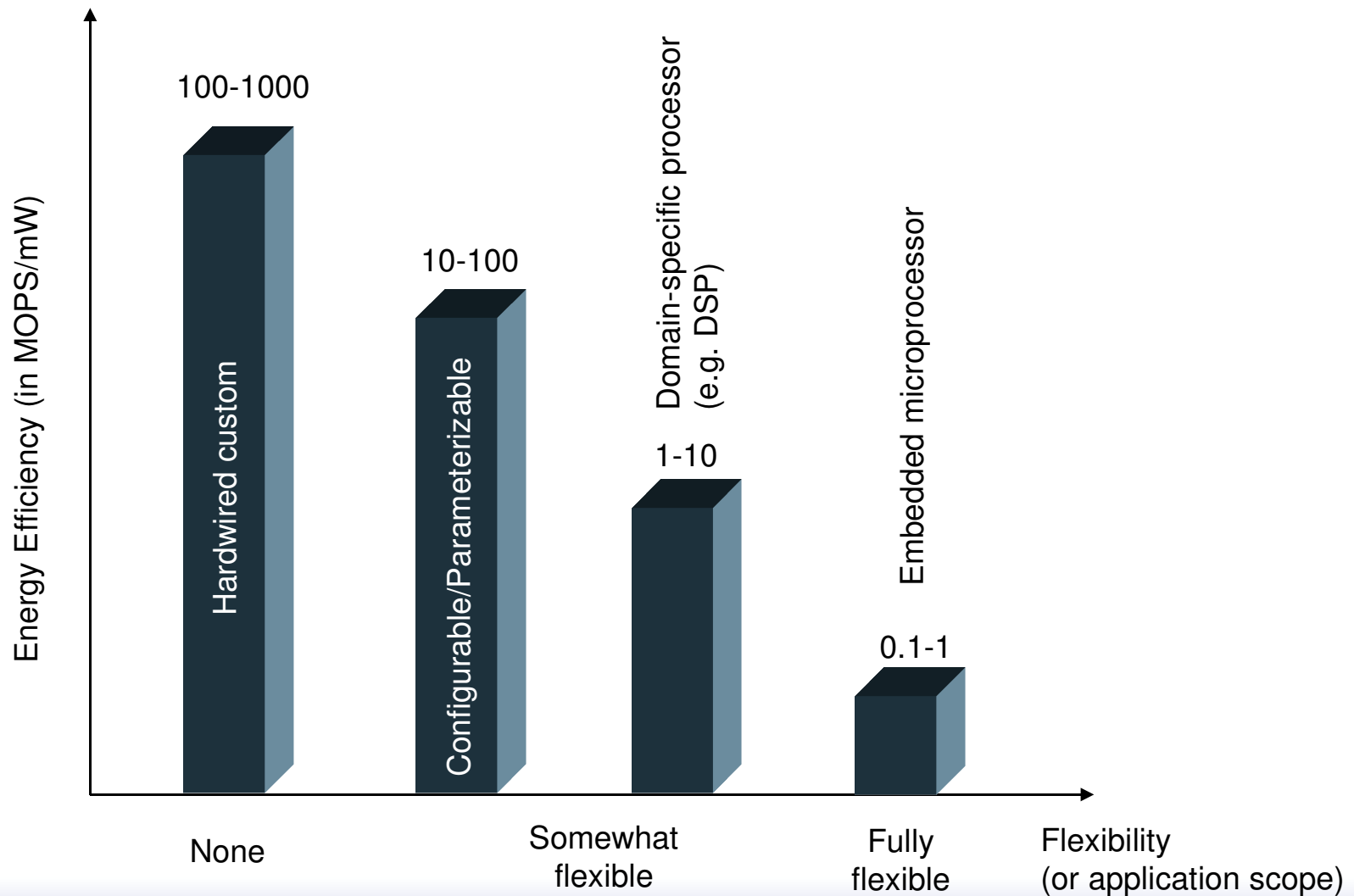


Rośnie przepaść złożonością “hardware-u” a wydajnością projektowania

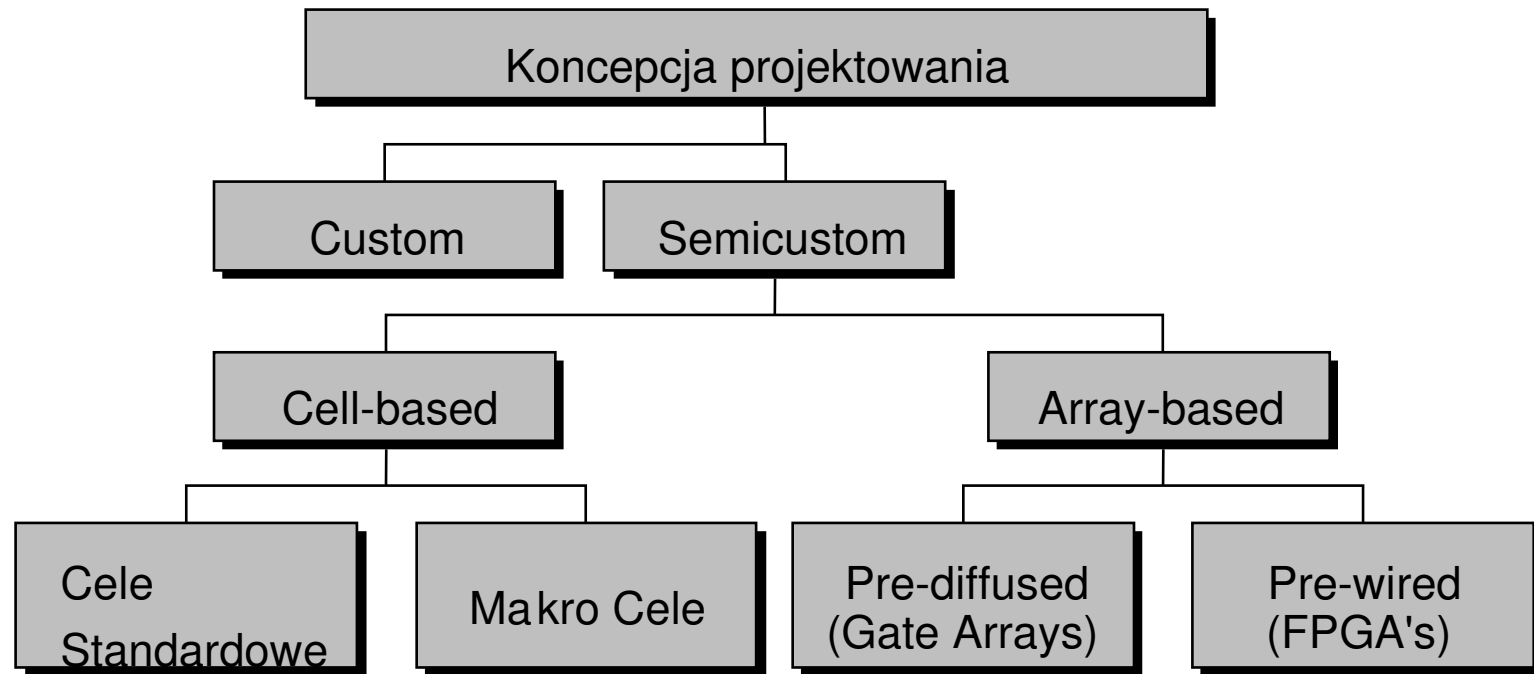
Prosty procesor



Wpływ rodzaju implementacji

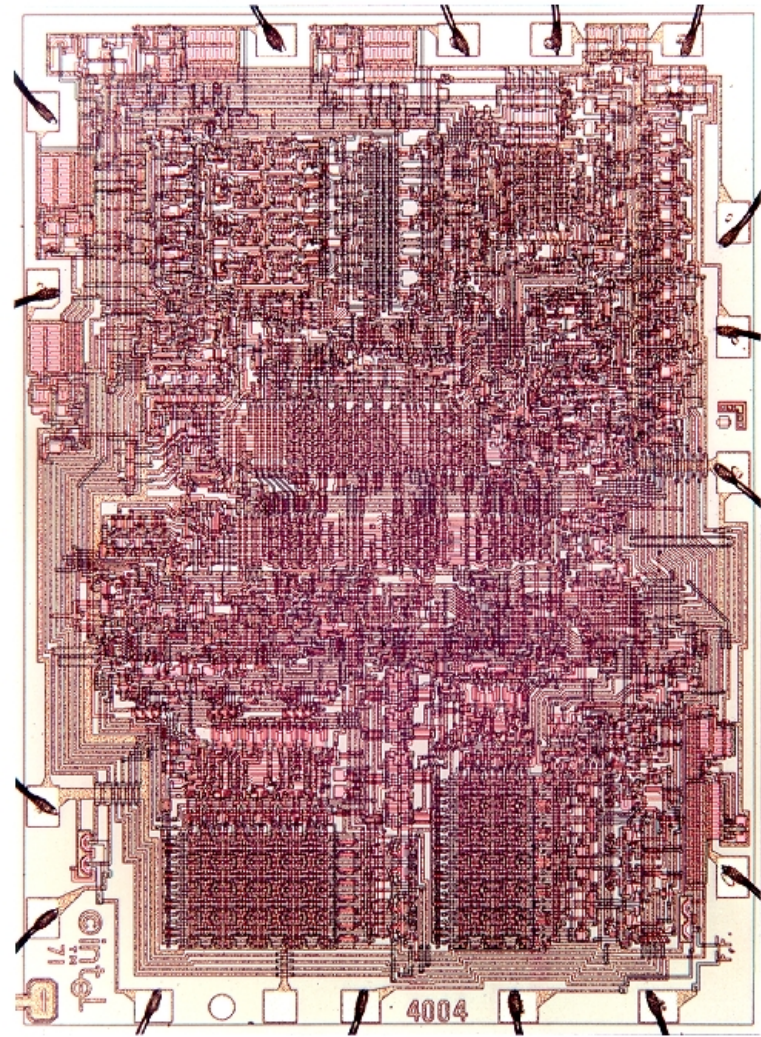


Projektowanie elektroniki cyfrowej (współczesne)



Istotne czynniki: szybkość, moc, koszt (opracowanie+produkcja), ilość, czas...

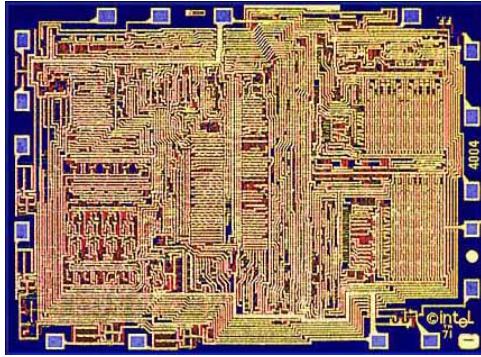
Projektowanie “Custom” (ręczne)



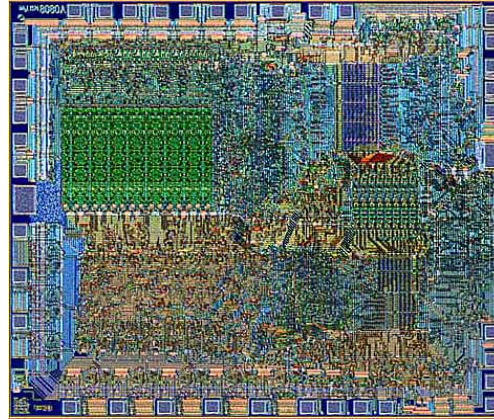
Intel 4004

Courtesy Intel

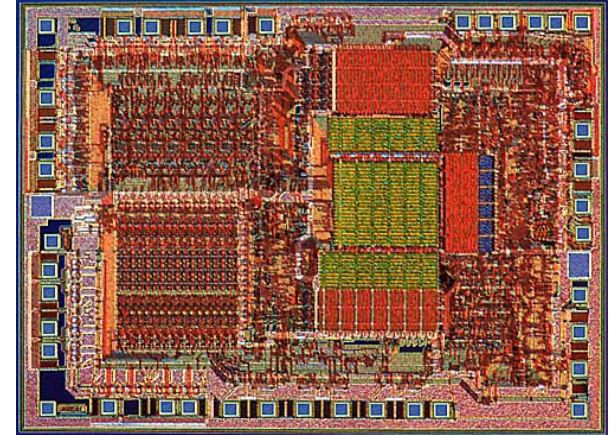
Przejs̄cie do automatycznego projektowania i regularnych bloków



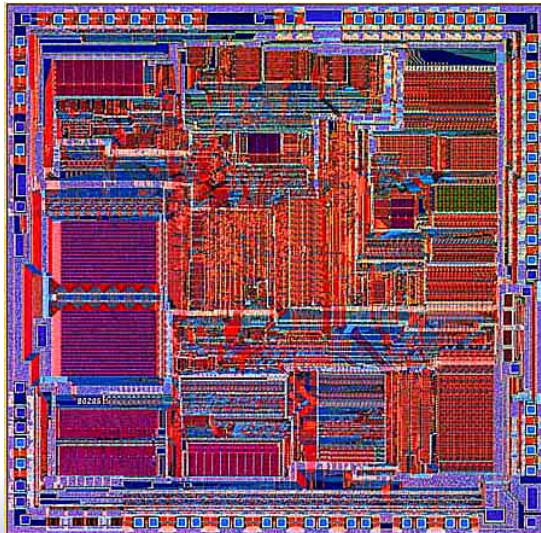
Intel 4004 ('71)



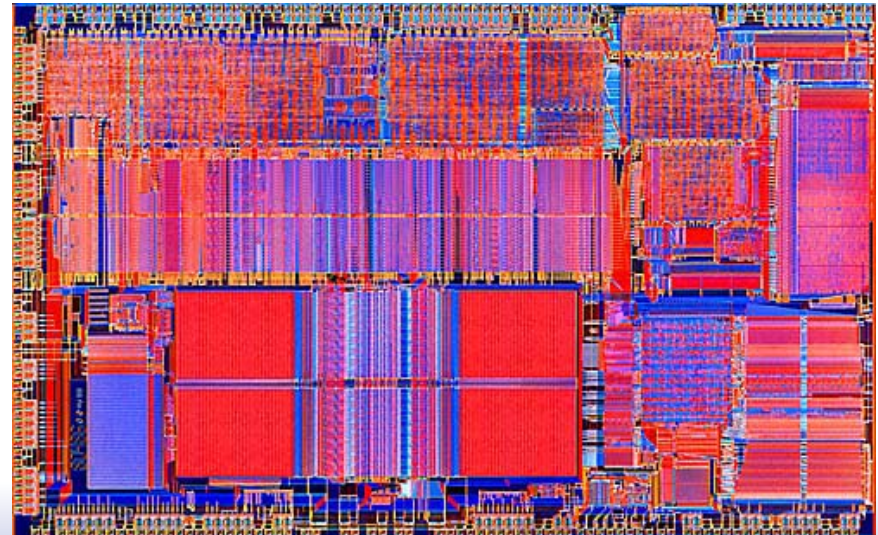
Intel 8080



Intel 8085



Intel 8286

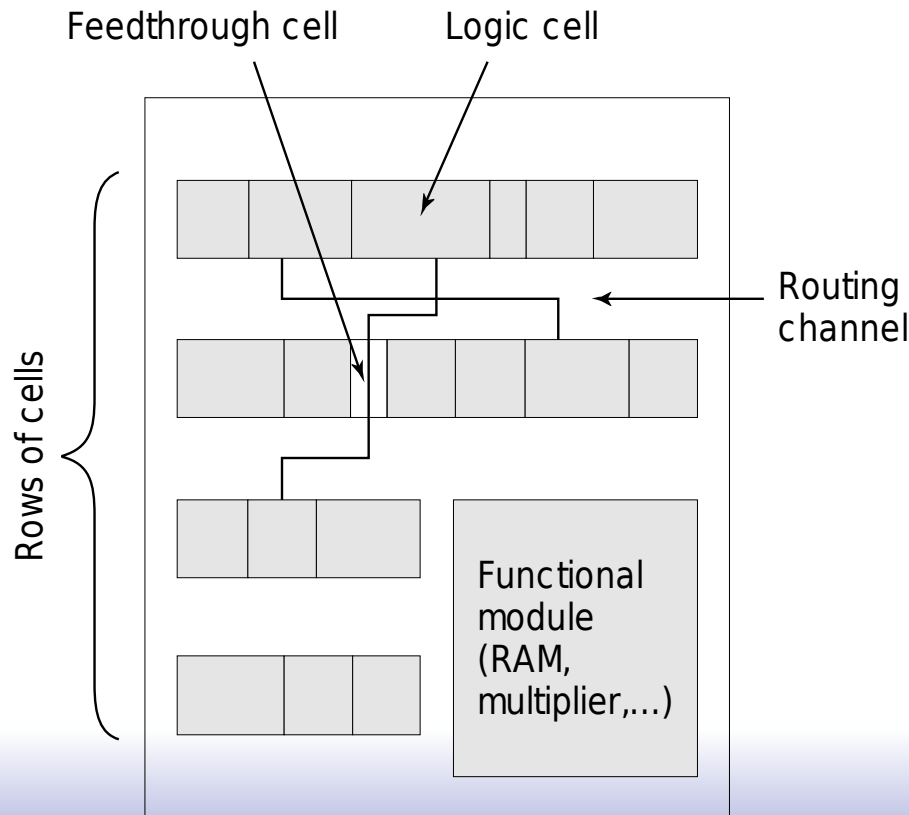


Intel 8486

Courtesy Intel

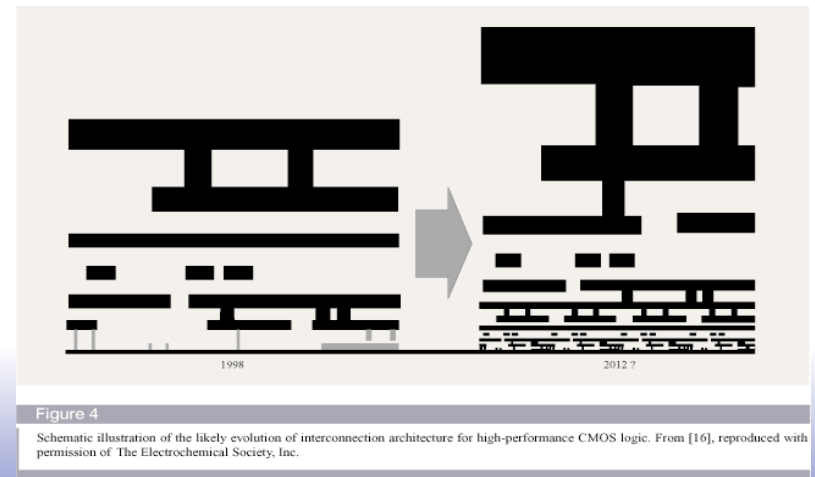
Projektowanie Cell-based (czyli cele standardowe)

Wcześniejszy standard

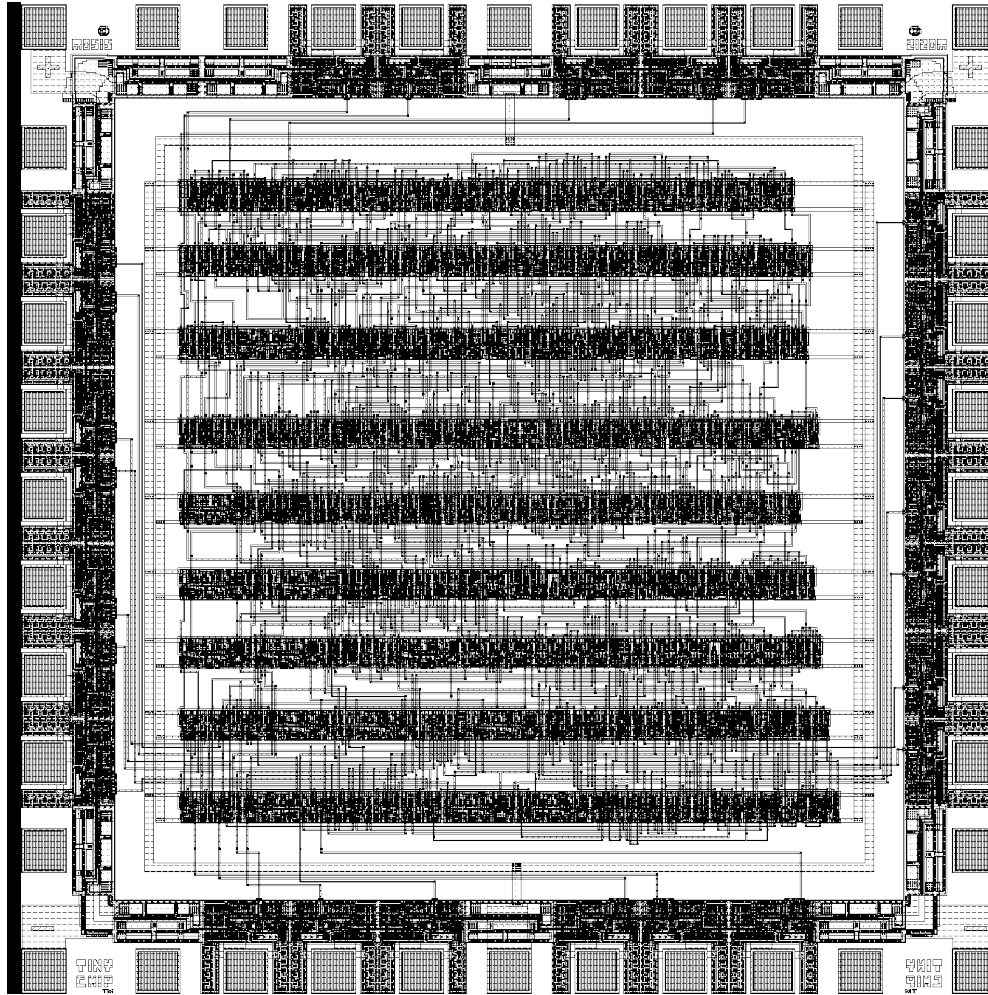


Obecne trendy

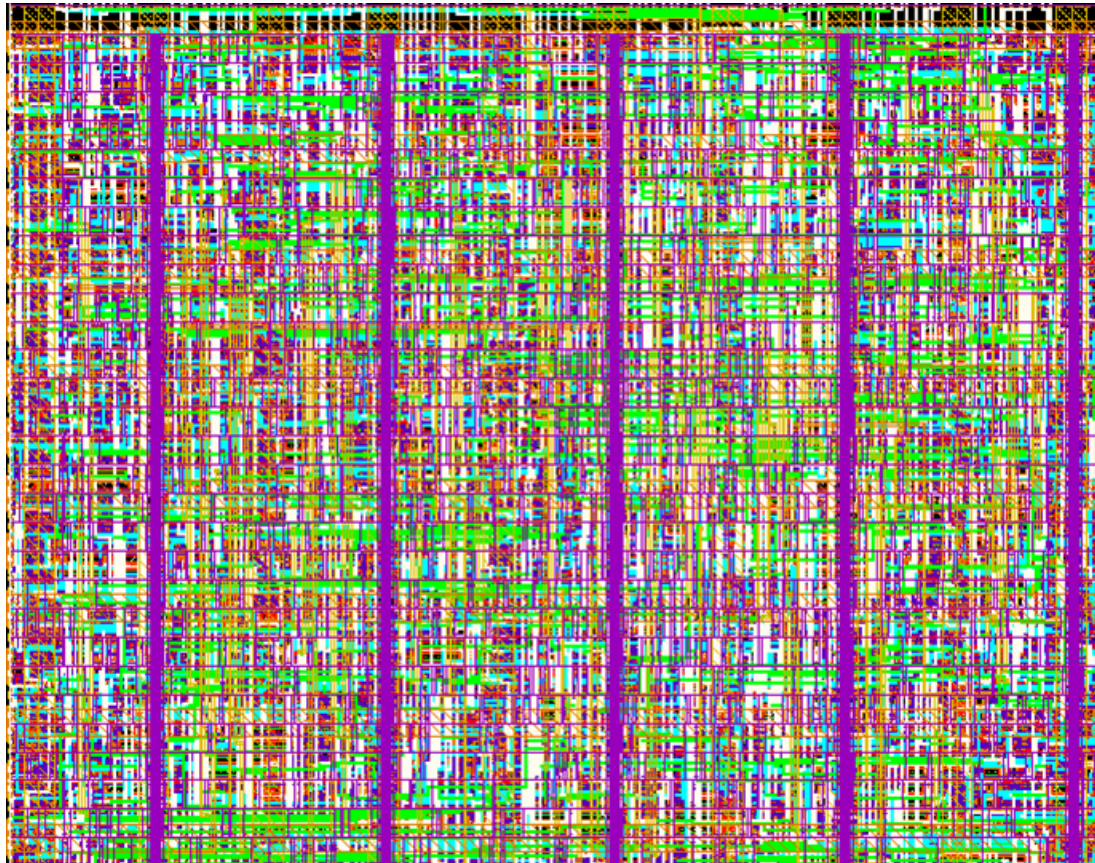
Wymaganie "Routing channel" redukuje się zwiększając ilość warstw metalu



Cele standardowe — jak było



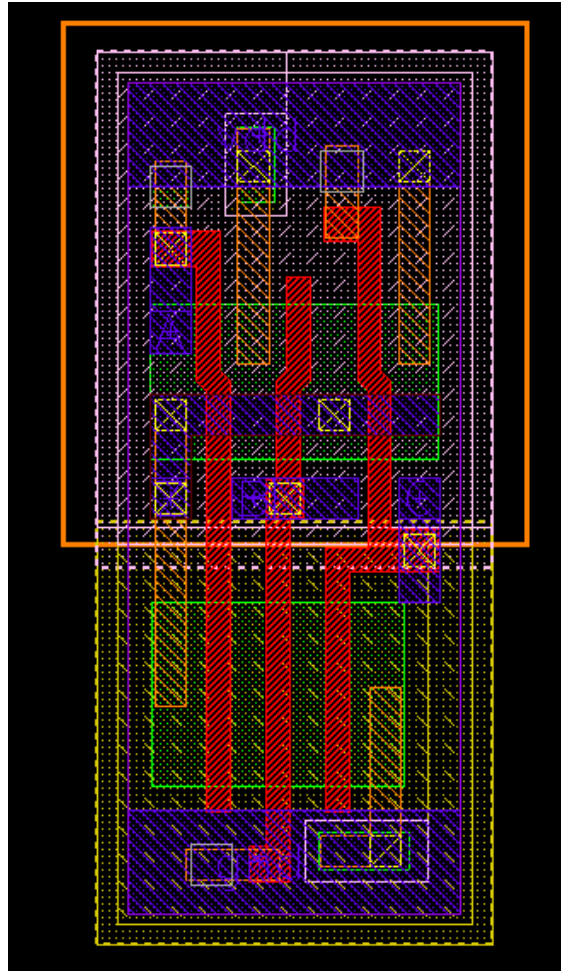
Cele standardowe – jak jest



*Struktura cel
ukryta pod
warstwami
połączeń*

*Gęstość
upakowania >90%*

Cela standardowa - przykład



Path	1.2V - 125°C	1.6V - 40°C
$In1-t_{pLH}$	$0.073+7.98C+0.317T$	$0.020+2.73C+0.253T$
$In1-t_{pHL}$	$0.069+8.43C+0.364T$	$0.018+2.14C+0.292T$
$In2-t_{pLH}$	$0.101+7.97C+0.318T$	$0.026+2.38C+0.255T$
$In2-t_{pHL}$	$0.097+8.42C+0.325T$	$0.023+2.14C+0.269T$
$In3-t_{pLH}$	$0.120+8.00C+0.318T$	$0.031+2.37C+0.258T$
$In3-t_{pHL}$	$0.110+8.41C+0.280T$	$0.027+2.15C+0.223T$

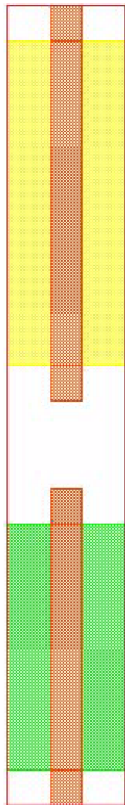
Bramka NAND 3-wejściowa

(ST Microelectronics):

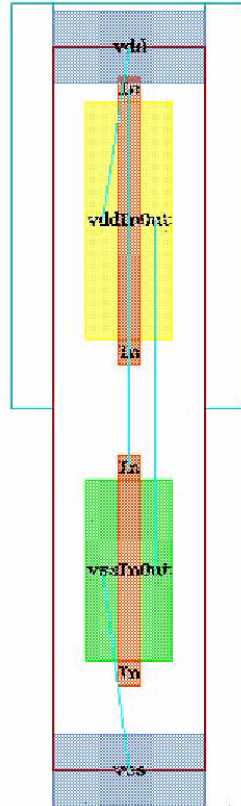
C = Load capacitance

T = input rise/fall time

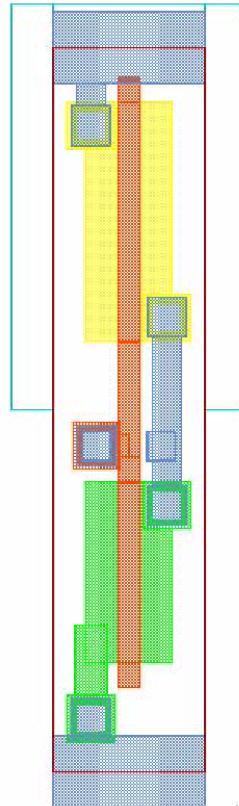
Automatyczna generacja cel



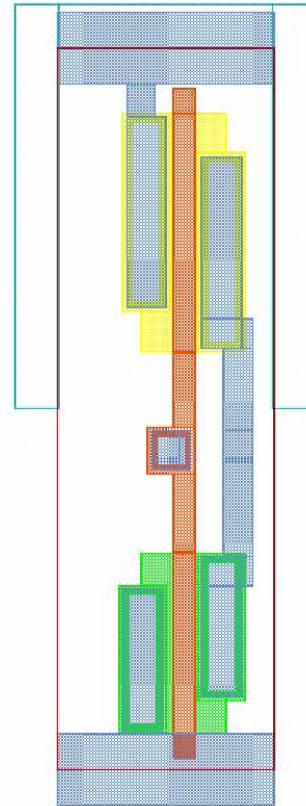
Initial transistor geometries



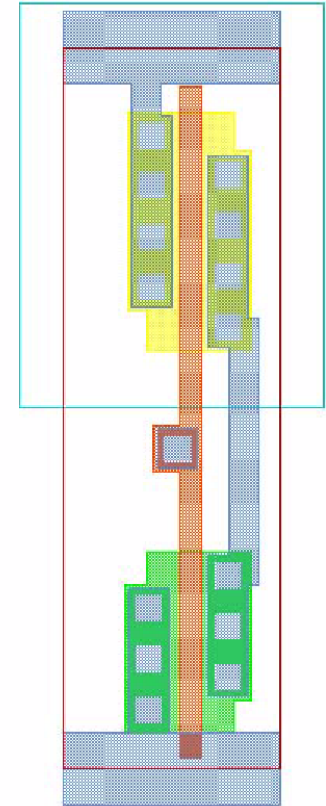
Placed transistors



Routed cell

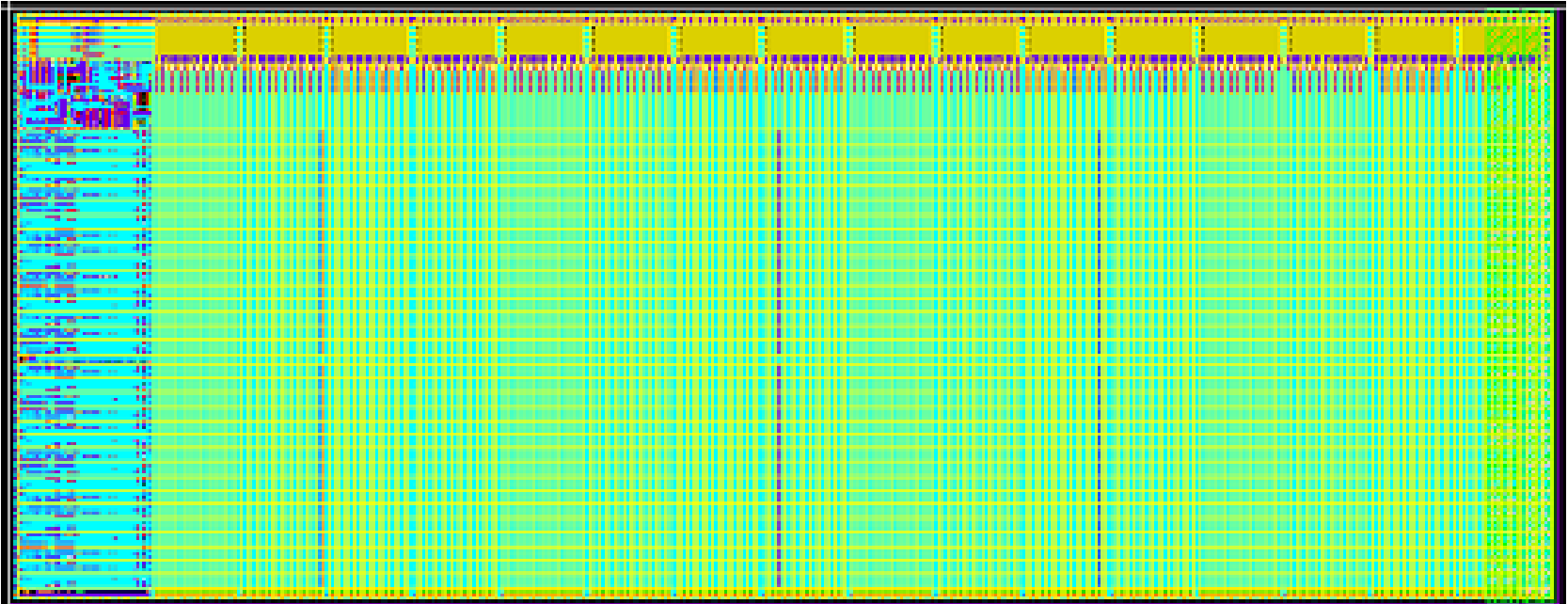


Compacted cell



Finished cell

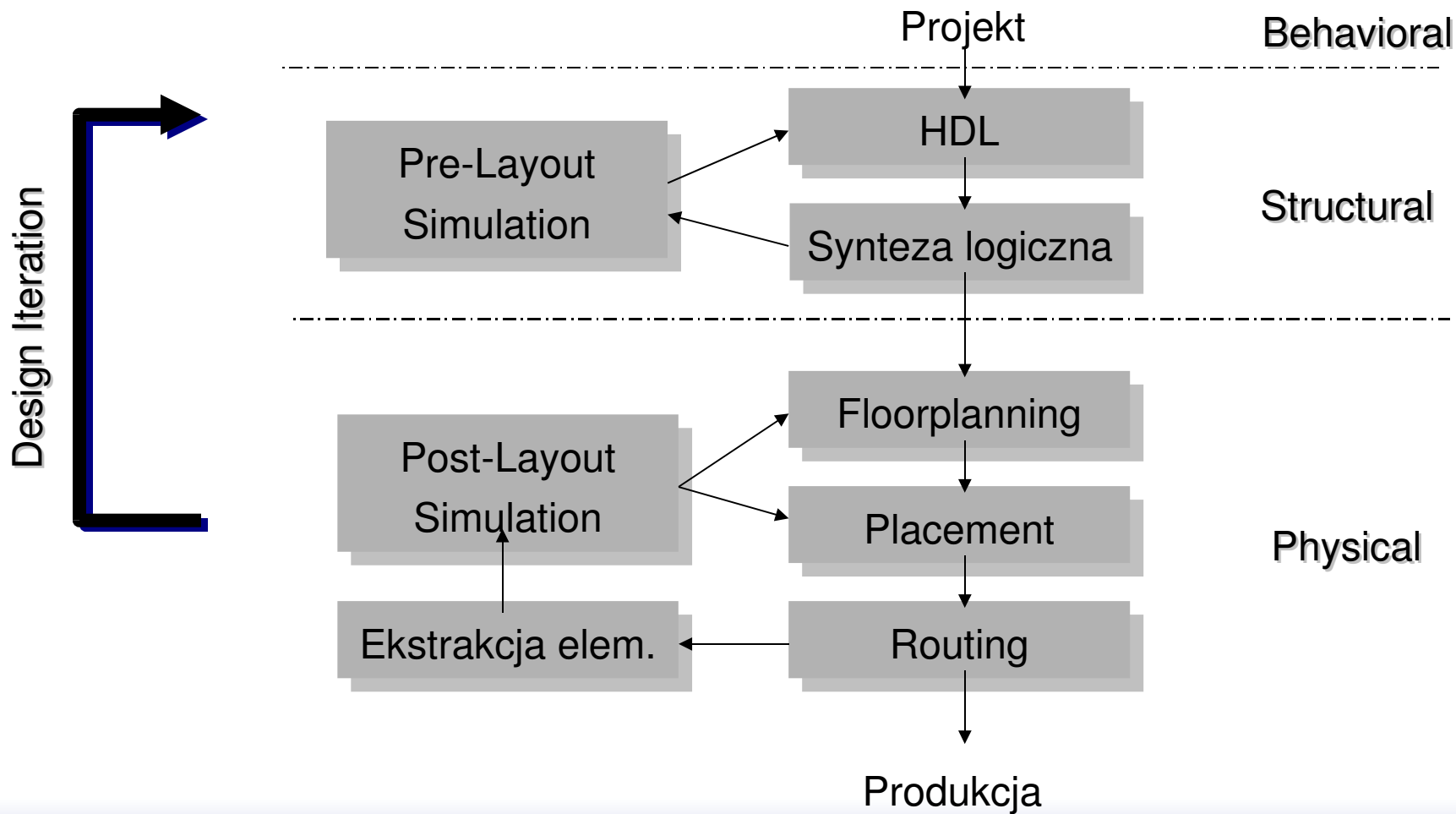
Makrocele i mikromoduły...



Przykład makroceli: 256×32 (or 8192 bit) SRAM

**Mikromoduły: Procesory DSP, interfejsy PCI, moduły FFT,
dekodery MPEG ...**

Projektowanie "Semicustom"



VHDL - przykłady

entity nand2 is

port (

a,b : in std_logic;

c : out std_logic

);

end nand2;

architecture logic of nand2 is

begin

c <= not (a and b);

end logic;

```
entity latch_D is
```

```
port(
```

```
    CLK      : in   STD_LOGIC;
```

```
    RESET    : in   STD_LOGIC;
```

```
    DIN      : in   STD_LOGIC;
```

```
    DOUT     : out  STD_LOGIC
```

```
);
```

```
end latch_D;
```

```
architecture behavior of latch_D is
```

```
begin
```

```
    dffar: process (CLK, RESET)
```

```
    begin
```

```
        if RESET='1' then
```

```
            DOUT <= '0';
```

```
        elsif rising_edge(CLK) then
```

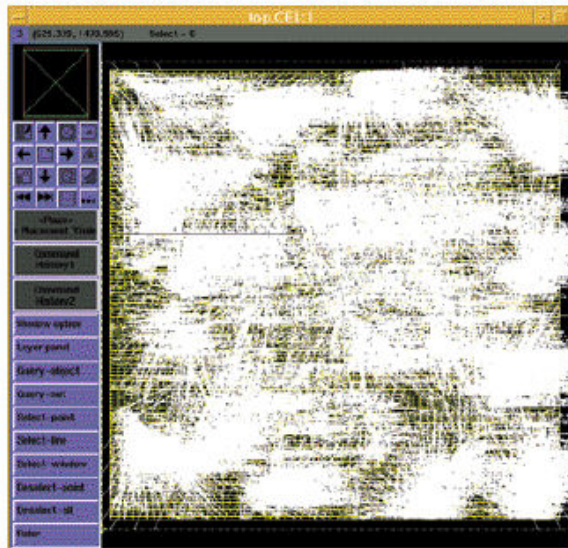
```
            DOUT <= DIN;
```

```
        end if;
```

```
    end process;
```

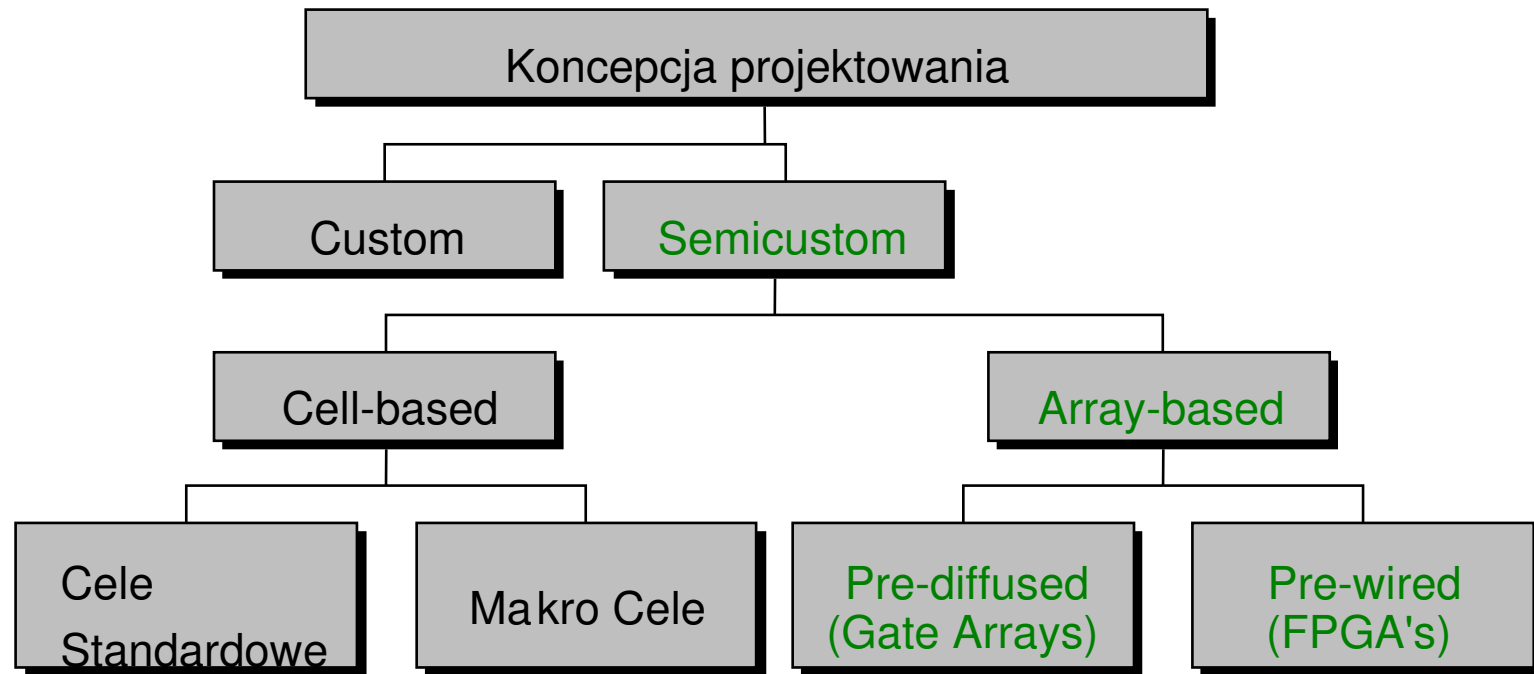
```
end behavior;
```

Problem “Design Closure”



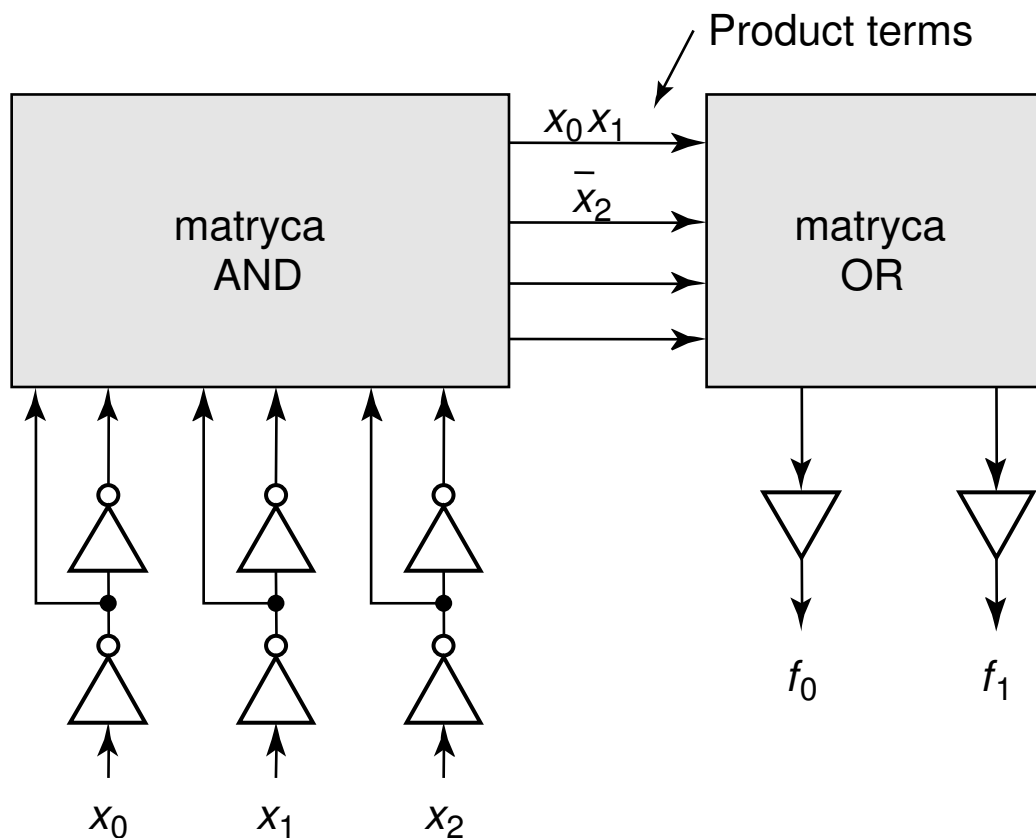
Iteracyjne usuwanie błędów czasowych (białe linie)

Projektowanie elektroniki cyfrowej...



Istotne czynniki: czas, koszt przy niewielkiej ilości

Idea PLA (programowalne układy logiczne)



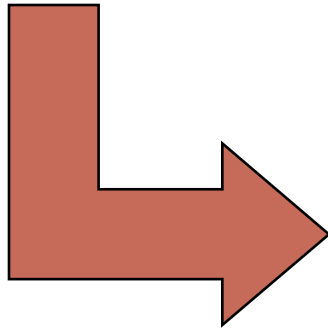
Logika dwupoziomowa

$$f_0 = x_0x_1 + \bar{x}_2$$

$$f_1 = x_0x_1x_2 + \bar{x}_2 + \bar{x}_0x_1$$

Każda funkcja logiczna może być wyrażona jako suma produktów typu (AND-OR)

minterm

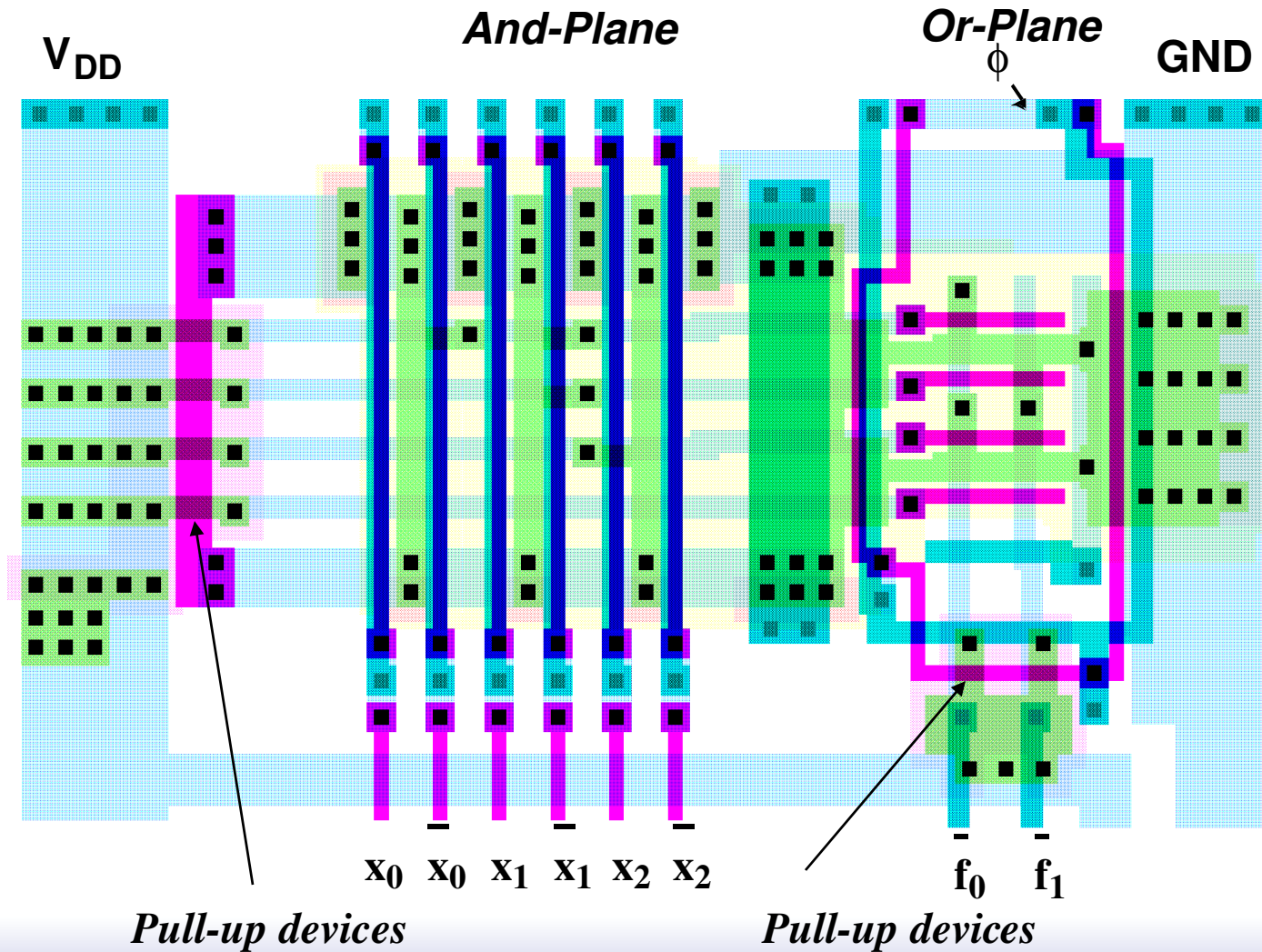


Inwertujący format (NOR-NOR) bardziej wydajny
(NAND-NAND wolniejszy)

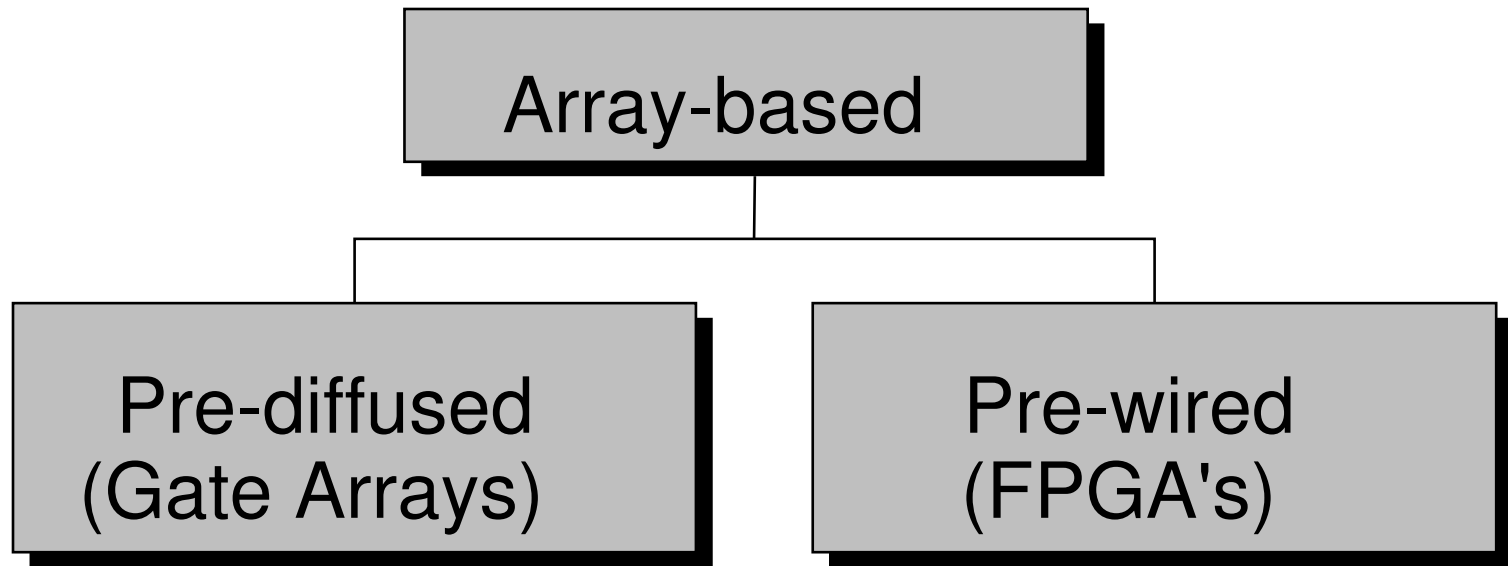
$$\bar{f}_0 = \overline{(\bar{x}_0 + \bar{x}_1) + \bar{x}_2}$$

$$\bar{f}_1 = \overline{(\bar{x}_0 + \bar{x}_1 + \bar{x}_2) + \bar{x}_2 + (x_0 + x_1)}$$

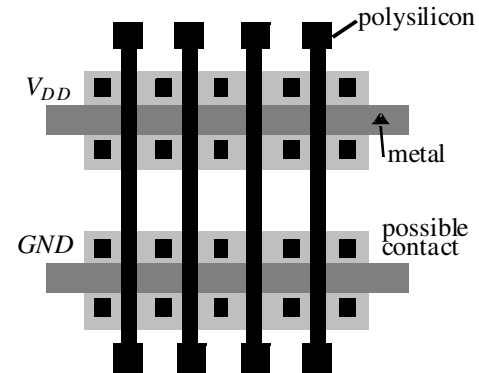
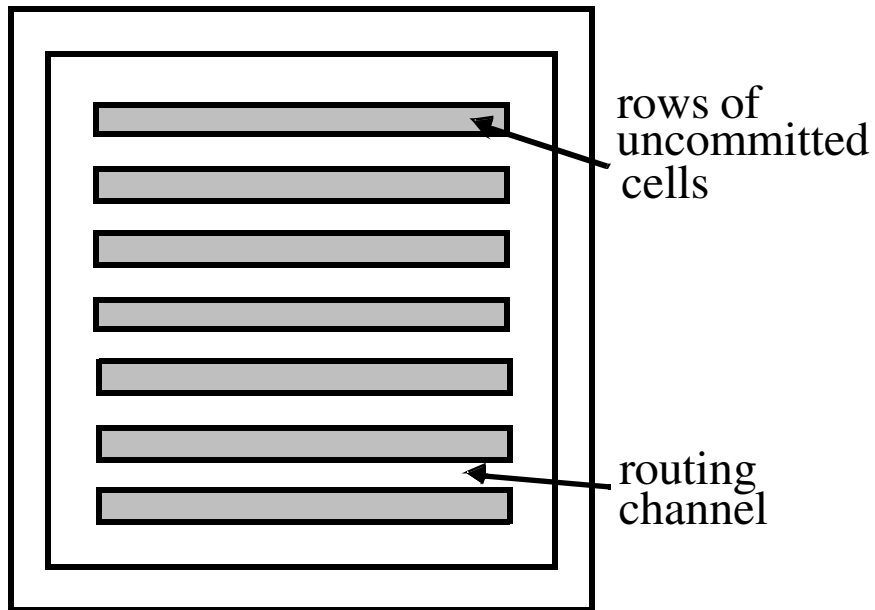
PLA Layout – przykład



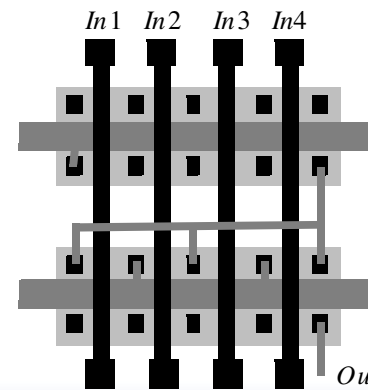
Projektowanie oparte na gotowych matrycach



Gate Array — Sea-of-gates



Cela pierwotna



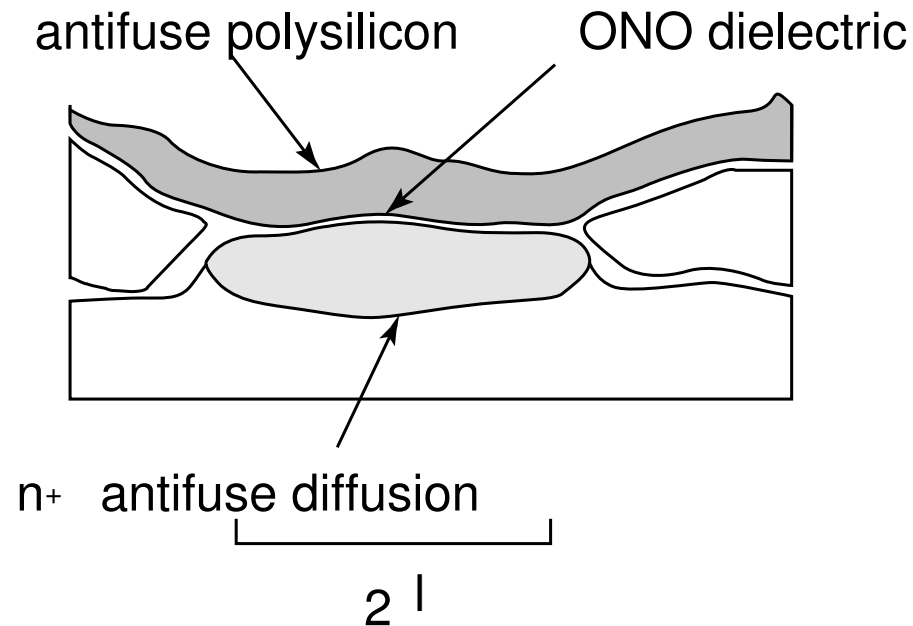
Cela wykonana
(4-input NOR)

Prewired Arrays

Klasyfikacja “prewired arrays” (field-programmable devices - FPGA):

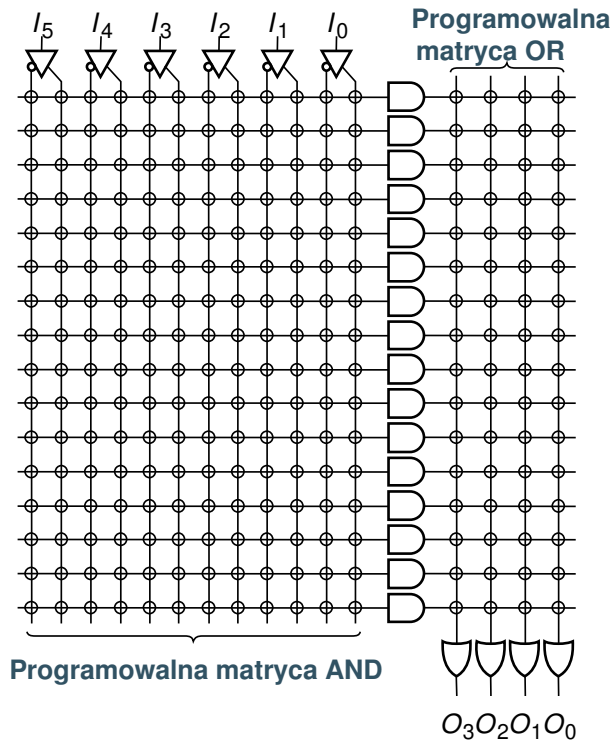
- Na podstawie typu pamięci
 - Fuse-based (programowane jednokrotnie)
 - Non-volatile (nie ulotne) oparte na EEPROM, Flash
 - RAM
- Typy logiki w układach programowalnych
 - Array-Based
 - Look-up Table
- Sposób wykonywania połączeń
 - Channel-routing
 - Mesh networks

Fuse-Based FPGA

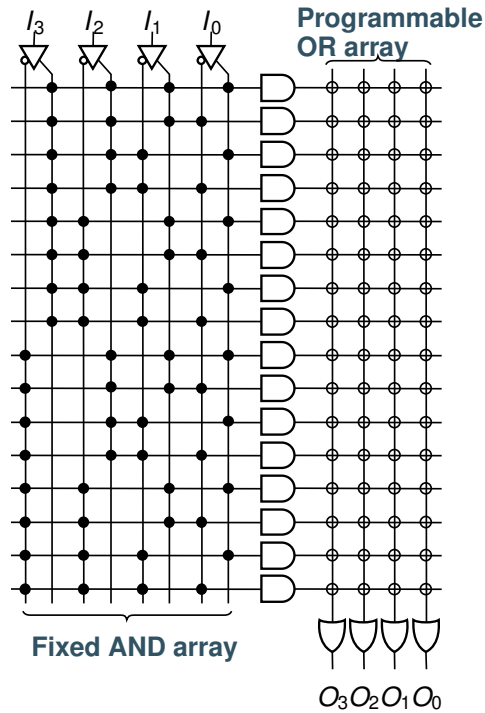


Wstępnie rozwarcie, zwarcie przez podanie impulsu prądowego

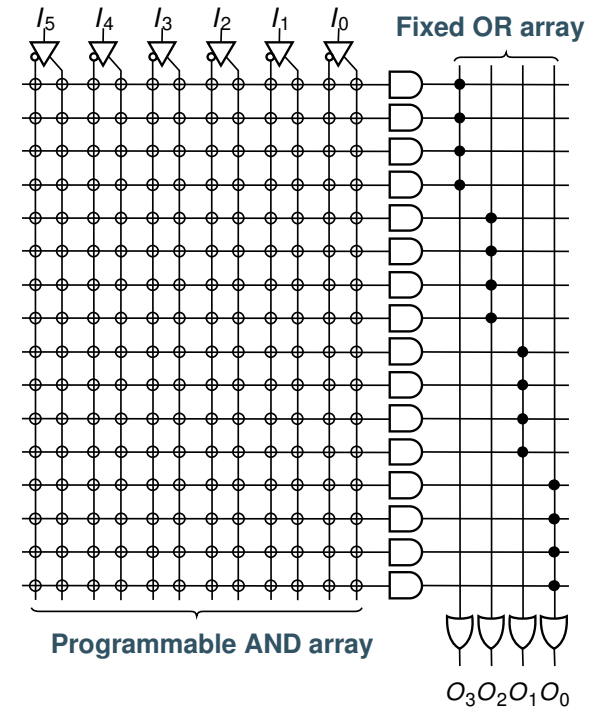
Logika programowalna "Array-Based", PLD



PLA



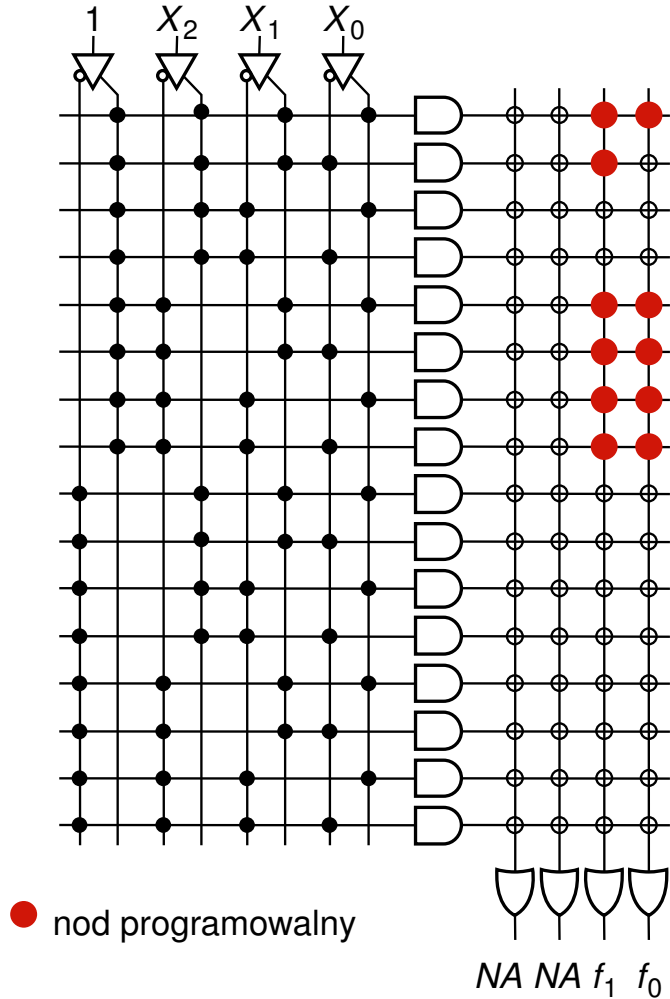
PROM



PAL

- ⊕ Oznacza połączenie programowalne
- Oznacza połączenie stałe

Programowanie PROM-u

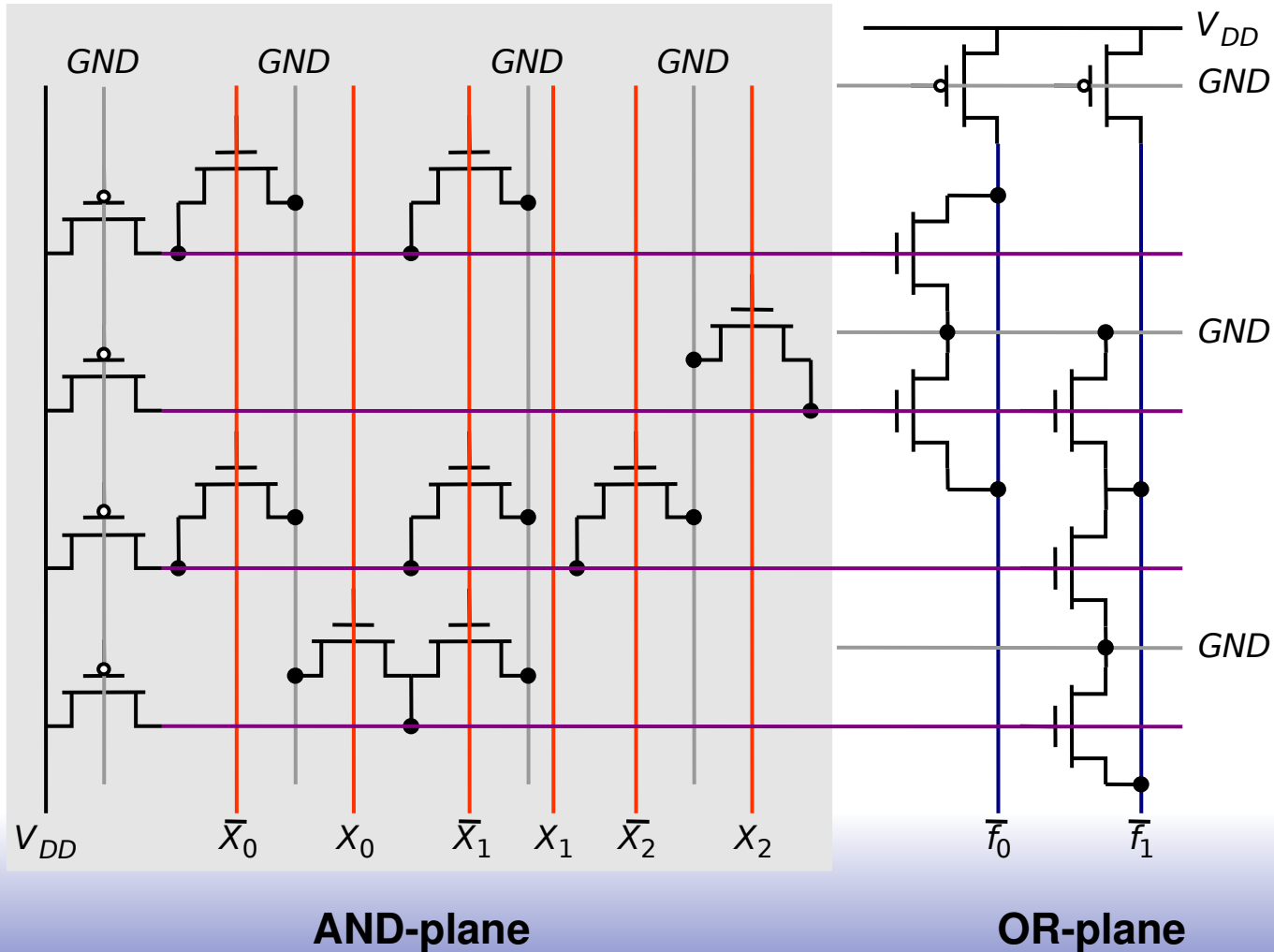


$$f_0 = x_0x_1 + \overline{x_2}$$

$$f_1 = x_0x_1x_2 + \overline{x_2} + \overline{x_0}x_1$$

PLA typu NOR-NOR

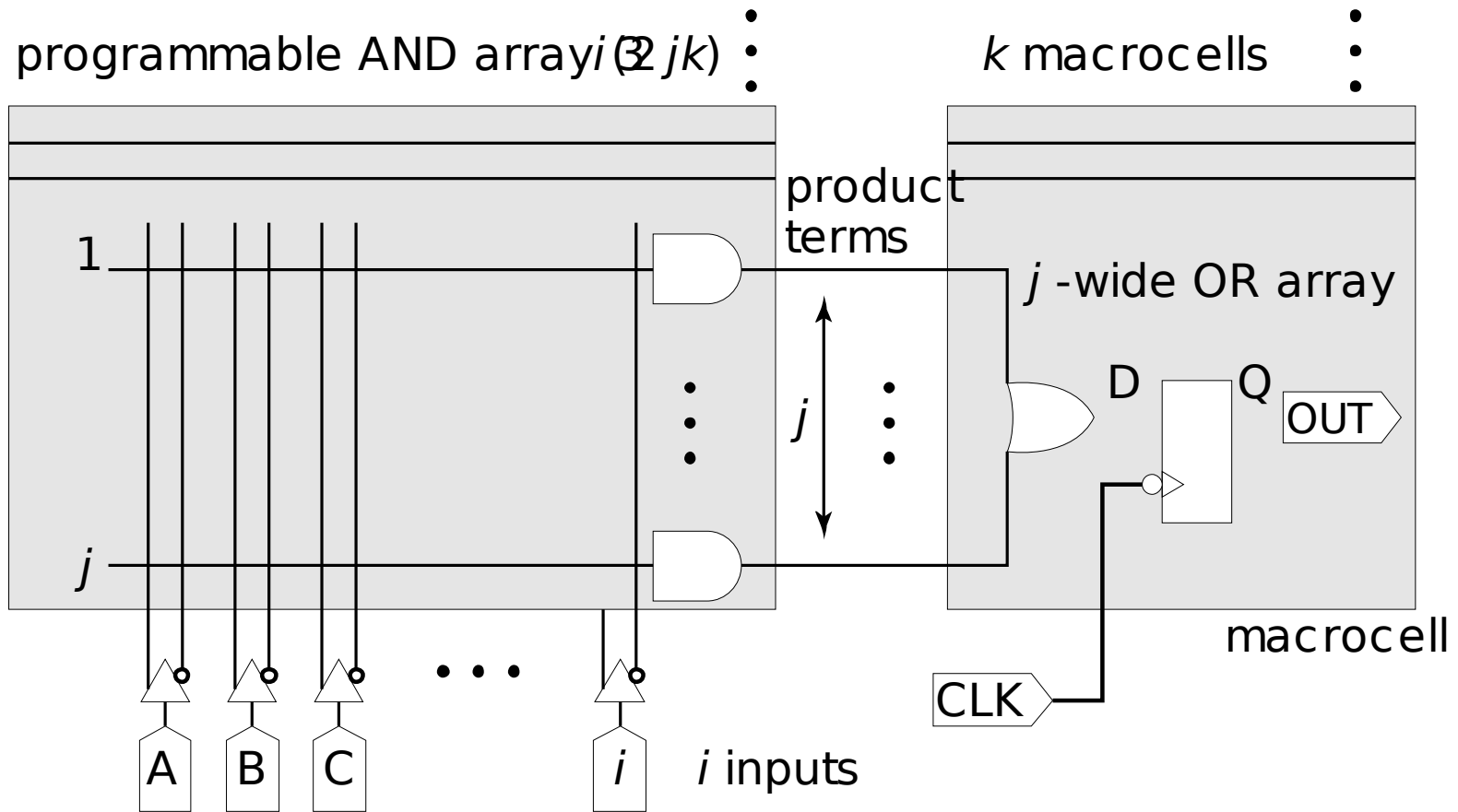
Pseudo-NMOS PLA



Ulepszenie PLA

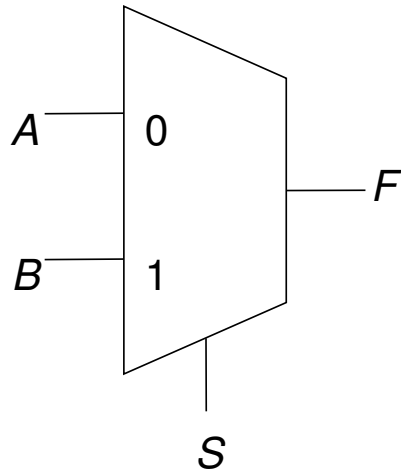
- ❑ Zamiast jednej dużej matrycy wiele małych (makrocele)
- ❑ Umożliwienie realizacji logiki sekwencyjnej – dodanie przerzutników

Bardziej złożony PAL - makrocela



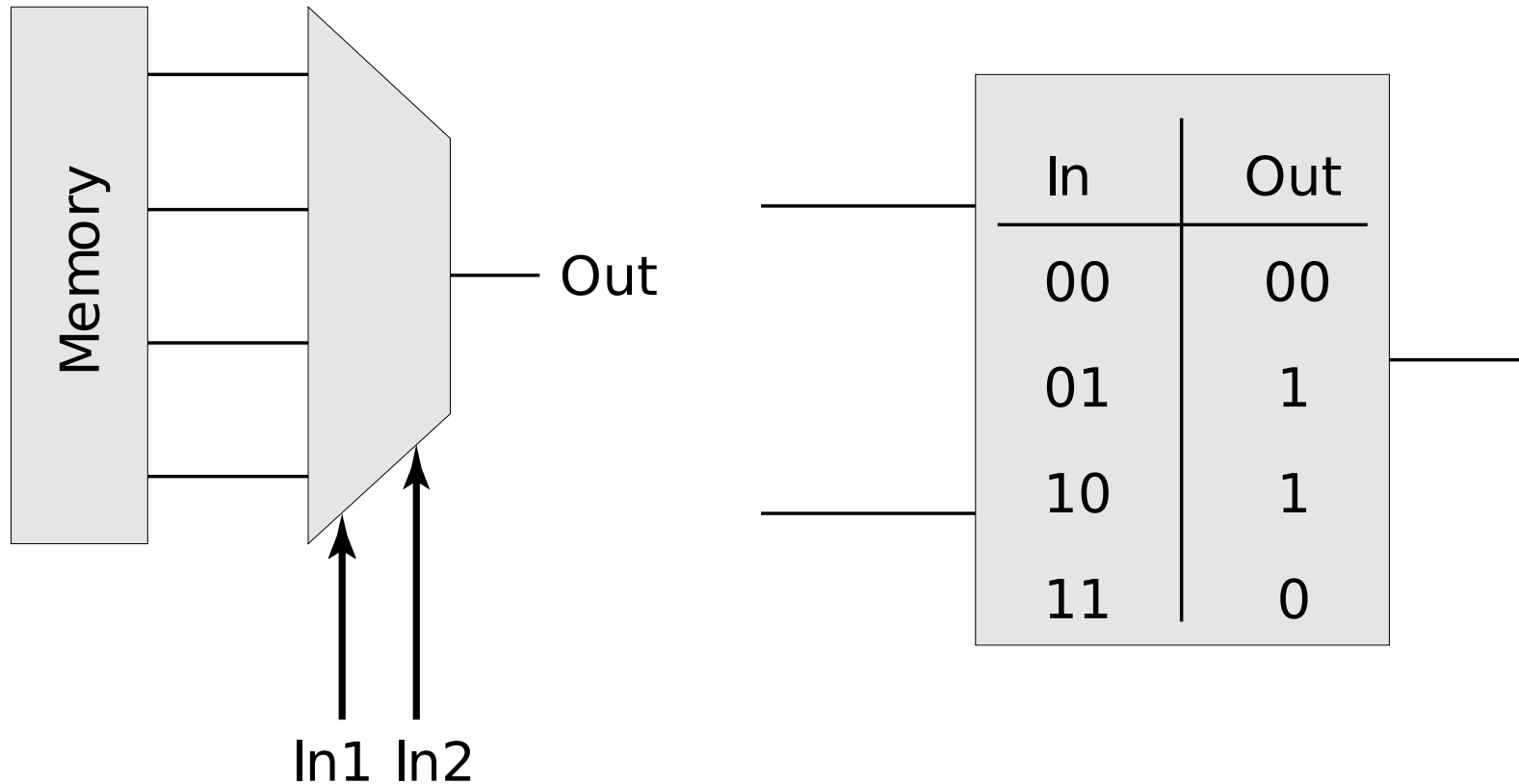
i inputs, j minterms/macrocell, k macrocells

Multiplexer jako programowalny blok logiczny

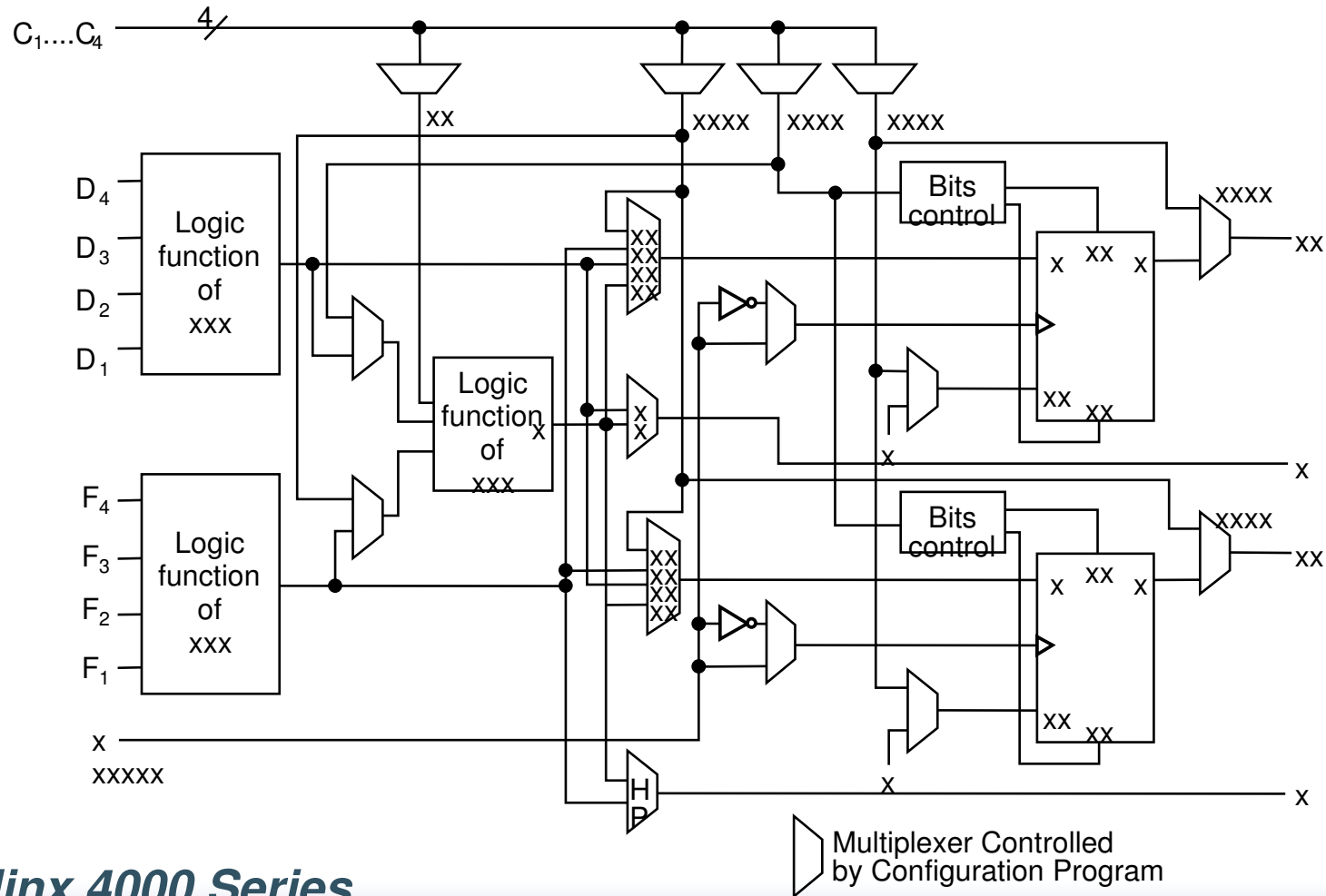


Configuration			
A	B	S	F=
0	0	0	0
0	X	1	X
0	Y	1	Y
0	Y	X	\overline{XY}
X	0	Y	\overline{XY}
Y	0	X	\overline{XY}
Y	1	X	$X \overline{1} Y$
1	0	X	\overline{X}
1	0	Y	Y
1	1	1	1

Cela logiczna typu Look-up Table (LUT)



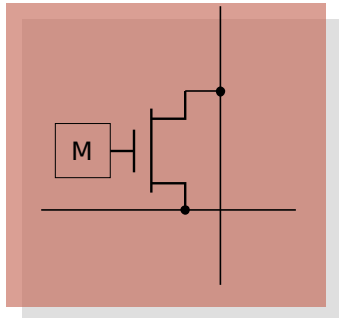
Cela logiczna LUT-Based



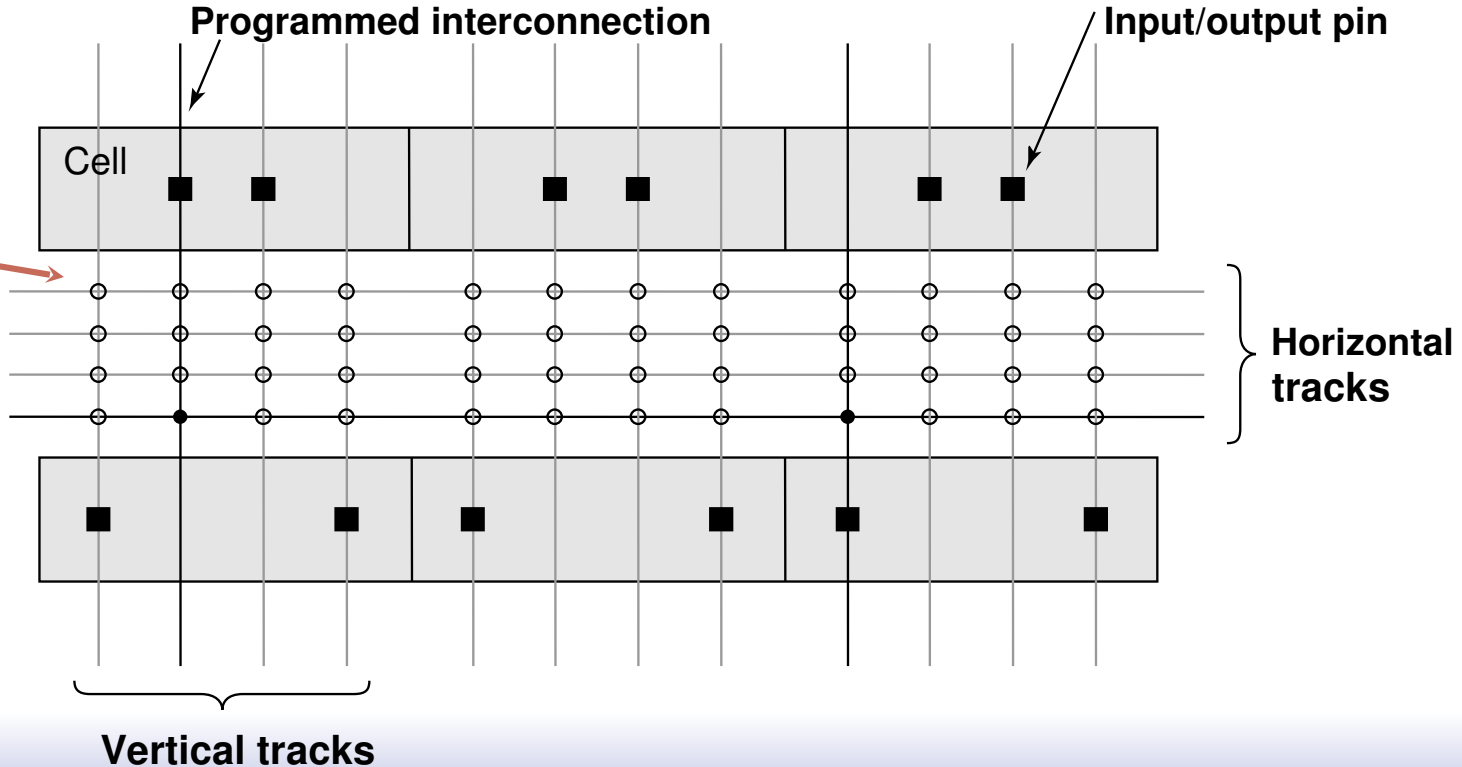
Xilinx 4000 Series

Courtesy Xilinx

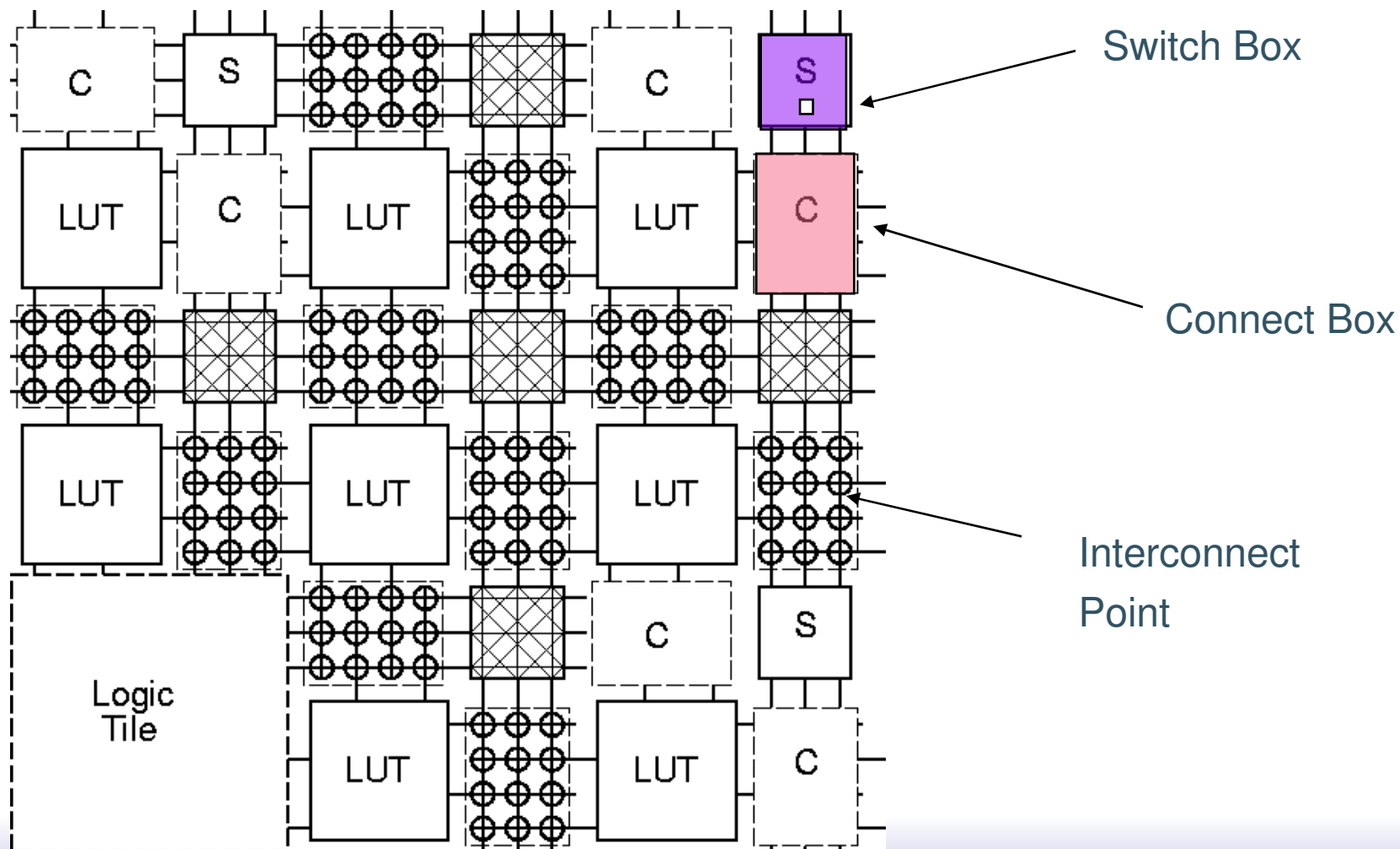
Programowalne połączenia Array-Based



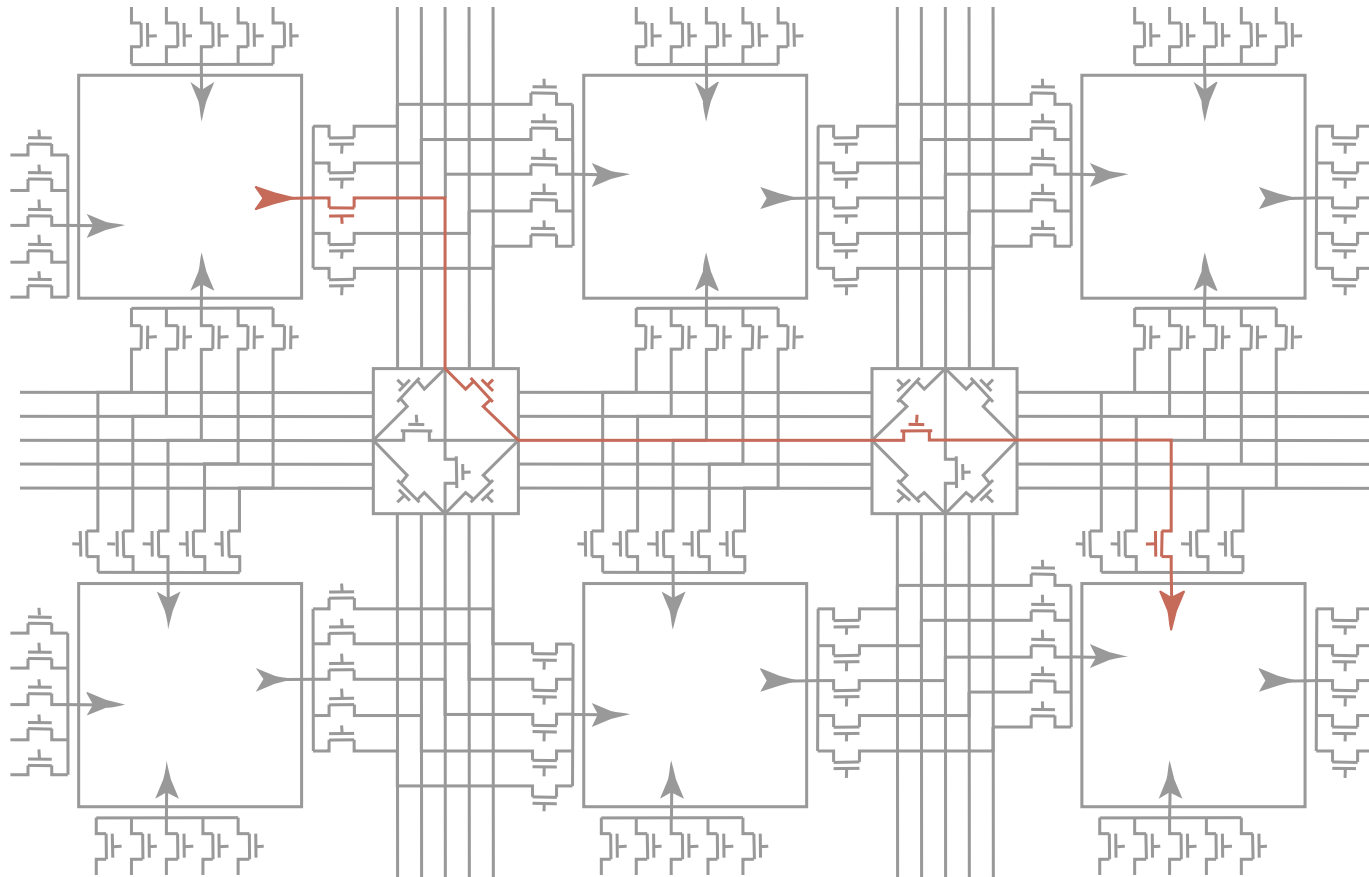
Interconnect Point



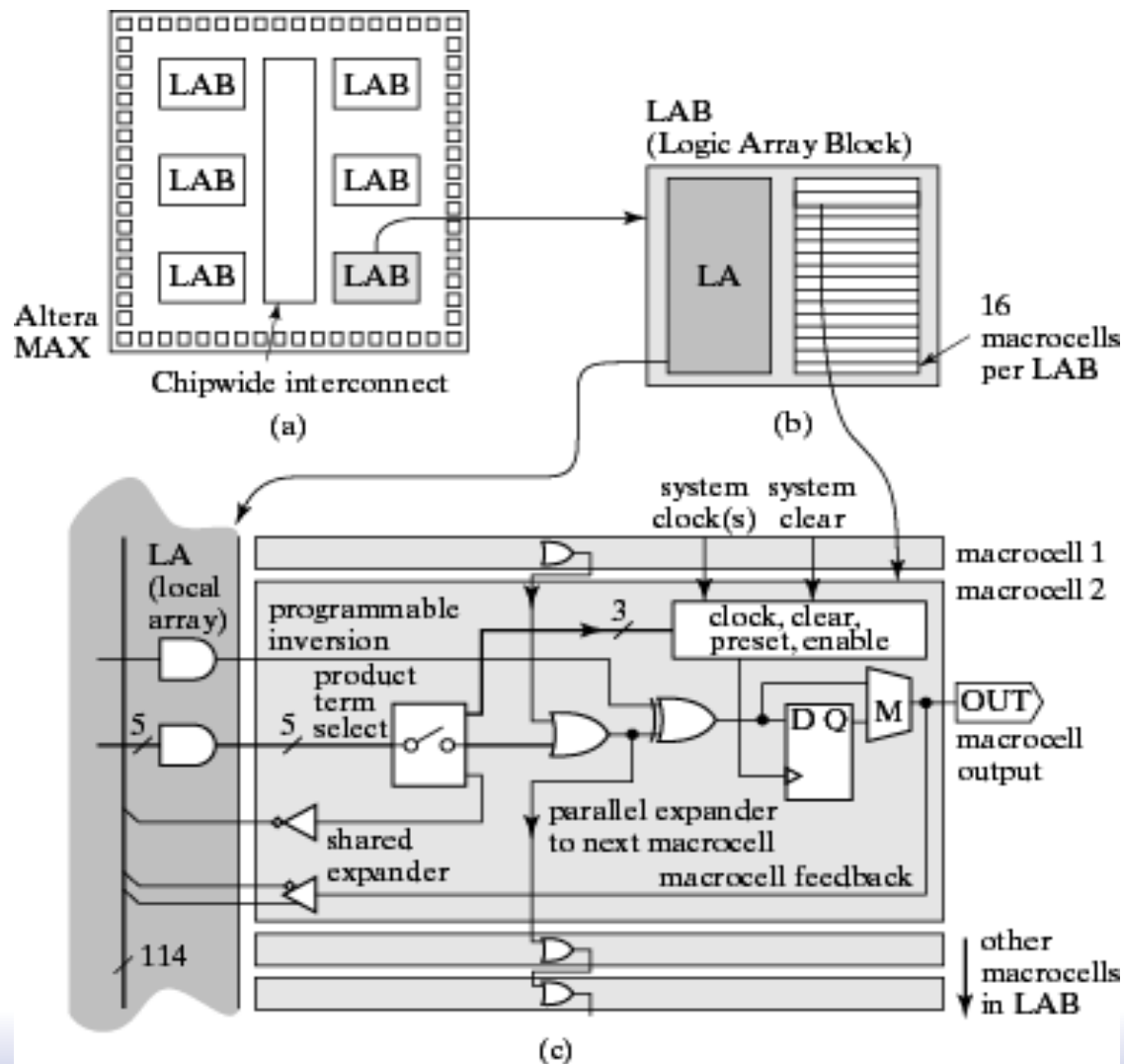
Sieć połączeń Mesh-based



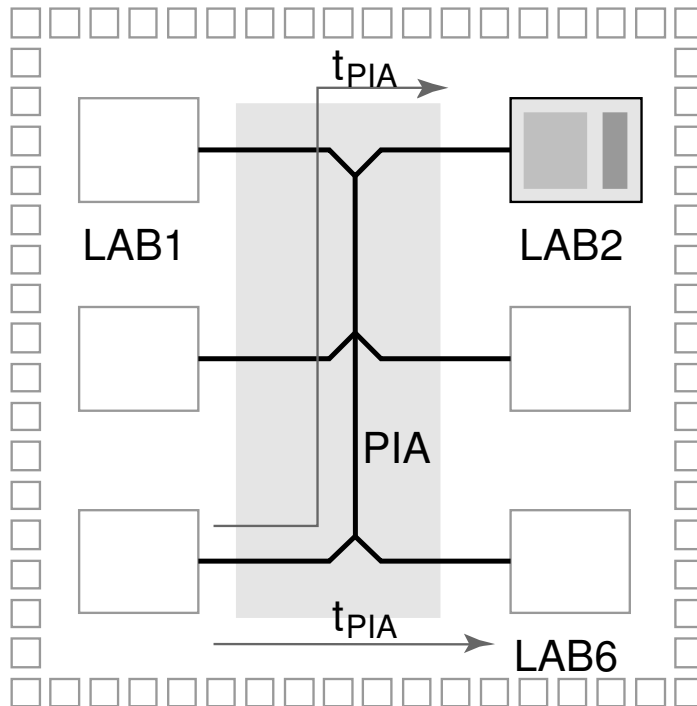
Implementacja tranzystorowa sieci Mesh



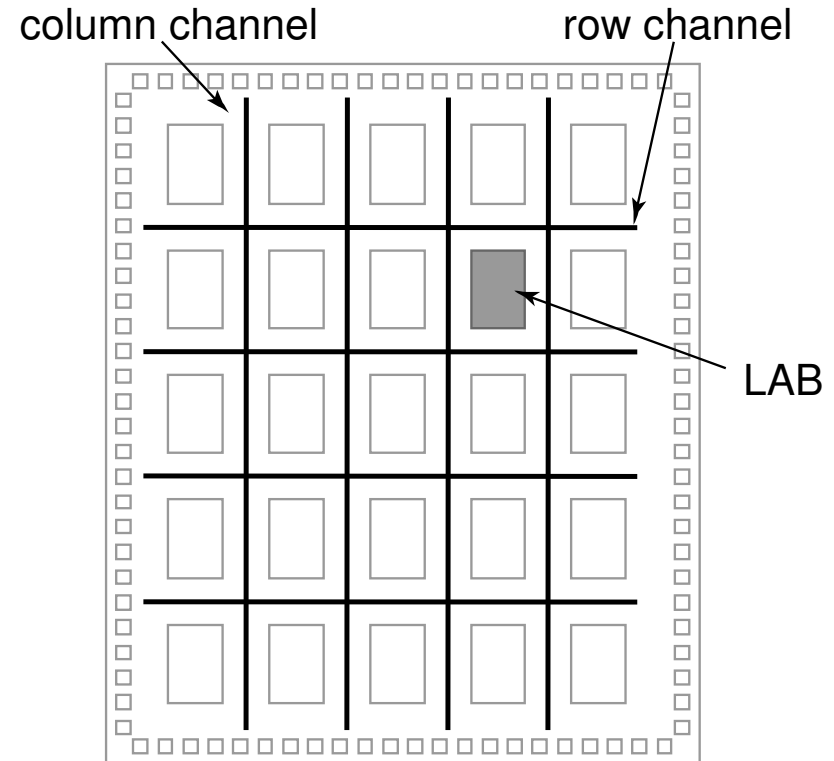
Rodzina Altera MAX



Architektury połączeń Altera MAX

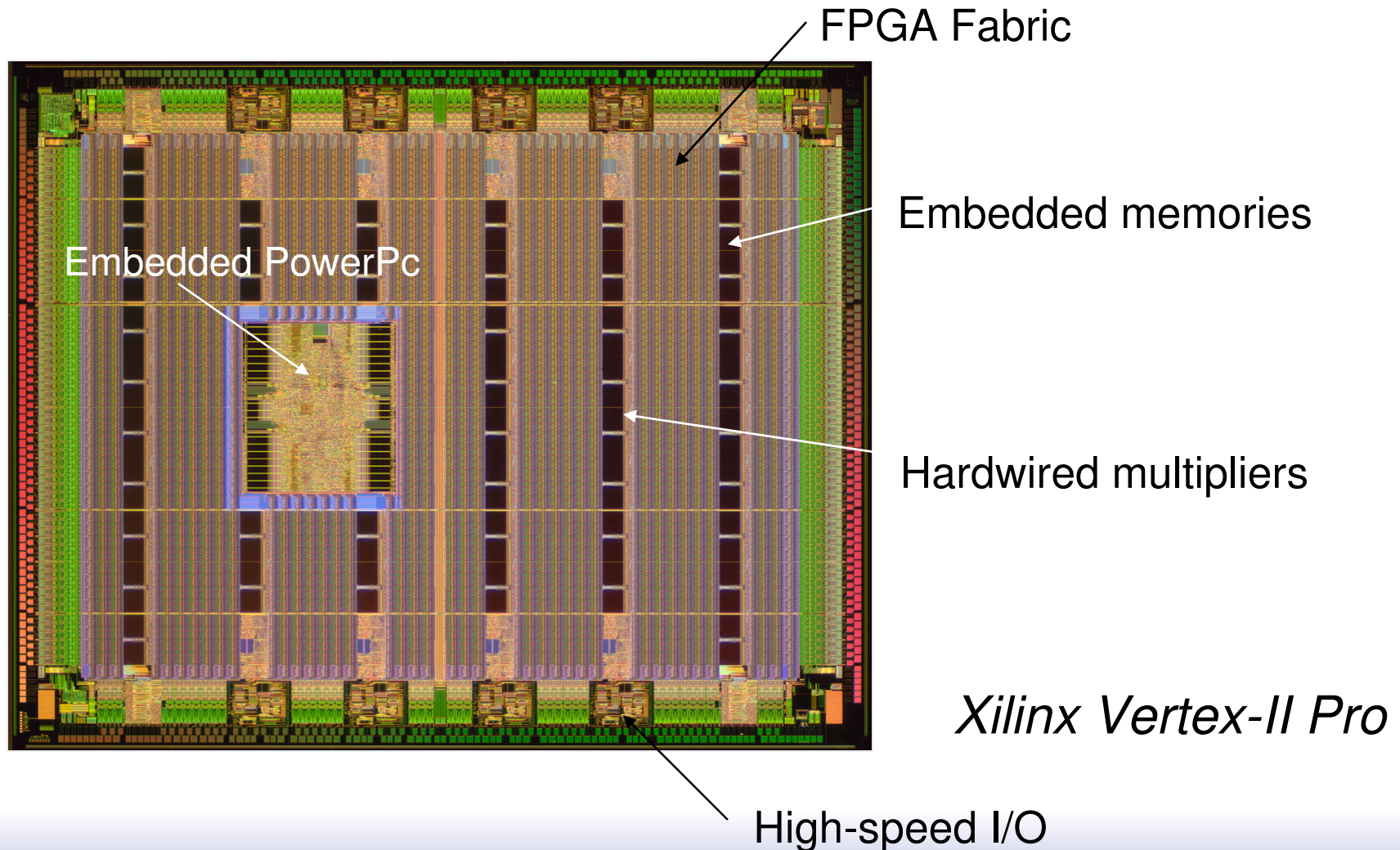


Array-based
(MAX 3000-7000)



Mesh-based
(MAX 9000)

Hybrydowe platformy programowalne



Courtesy Xilinx