LSI Logic

CoreWare reverse engineered Cells Library (used in derivative CW33300 PSXCPU at least)

(Such a primitive description so far. Also cells without any parameters like size, switching speed or parameters for SPICE. For our purposes this is not required)

Rev A2

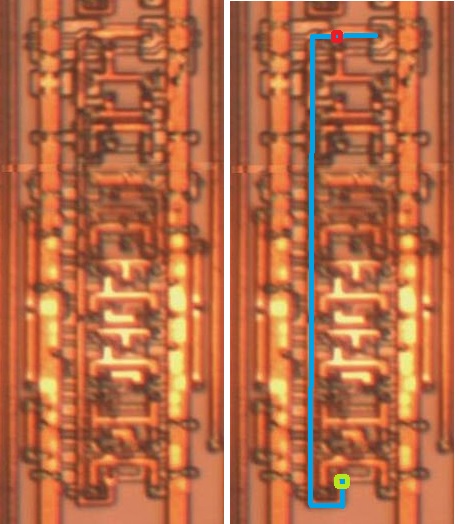
# Overview

CoreWare framework is based on standard cells.

Cells are built on CMOS technology. P and N channel transistors differ in size (P-doping is 1.3-1.5 times thicker). Diffusion is shown in yellow, and the polysilicon as purple. P-type diffusion is usually thickier rather diffusion of N-type, and is closer to the power supply. Accordingly, the diffusion of N-type is usually close to the ground.

Standard cells are placed by rows. Connections between cells are made using two layers of metal. (Channel Router).

Quite often automatic router place route directly on the area of the cell (via M1), if there exist free space for it:



# NOT

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| x | output |

x = ~a

Mini inverter.



# NOT1

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| x | output |

x = ~a

Nothing special.



# NOT2

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| x | output |

x = ~a

2 pairs of transistors



 Topologically very similar to NAND2X:



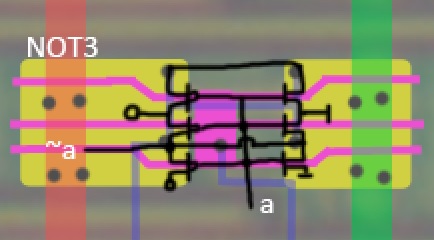
So be very careful not to mess up. The NAND2X is shorter than the NOT2, and also the «legs» on the right side go at different angles.

# NOT3

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| x | output |

x = ~a

3 pairs of transistors

# NOT4

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| x | output |

x = ~a

4 pairs of transistors

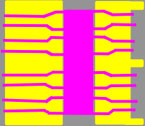
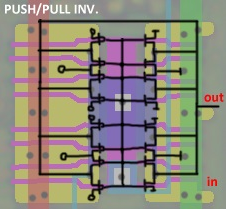


# NOT8

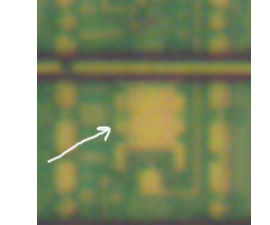
|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| x | output |

x = ~a

8 pairs of transistors

These inverters are easy to spot by the rather impressive metal shield (this is the output):

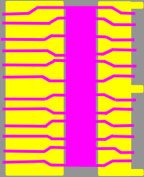


# NOT12

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| x | output |

x = ~a

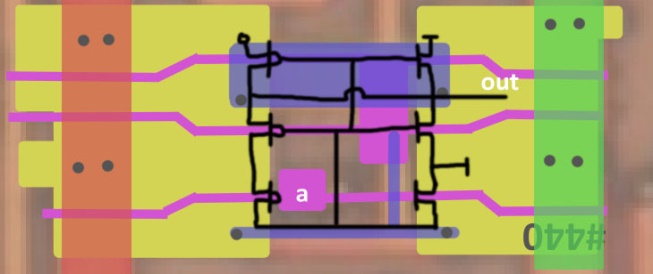
12 pairs of transistors



# BUF2X

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| x | output |

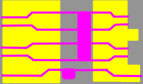
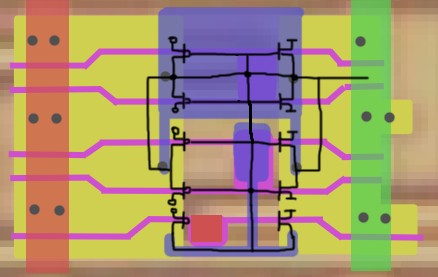
x = a

# BUF3X

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| x | output |

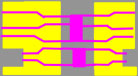
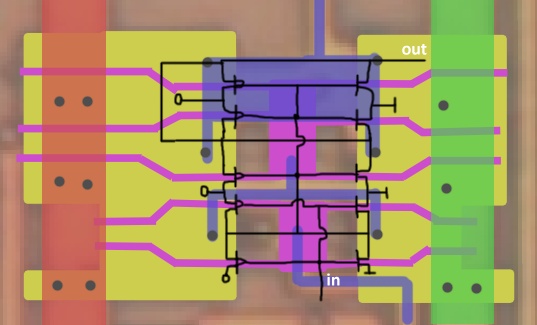
x = a

# BUF3X2

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| x | output |

x = a

# BUF4X

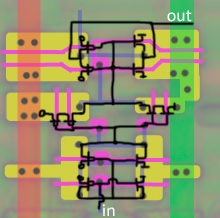
|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| x | output |

x = a

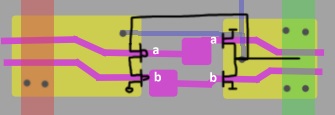
Double buffer.

The double buffer differs from the regular buffer in that it does out = not(not(not(in)))), while the regular buffer does just out = not(not(in)).

Hence the propagation delay of the double buffer will be longer than that of the regular buffer.

 The top of this cell is very similar to NOR2X:

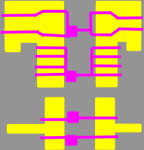
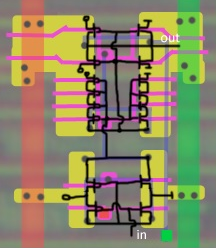


# BUF5X

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| x | output |

x = a

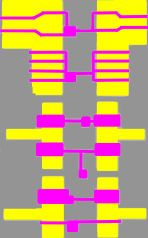
Double buffer, 4 pairs of transistors

# BUF6X

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| x | output |

x = a



# AND1

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| x | output |

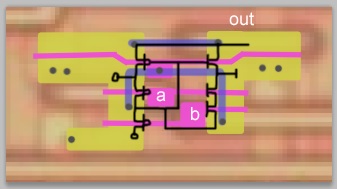
x = a & b



# AND

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| x | output |

x = a & b

# AND2X

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| x | output |

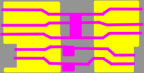
x = a & b



# AND3X

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| x | output |

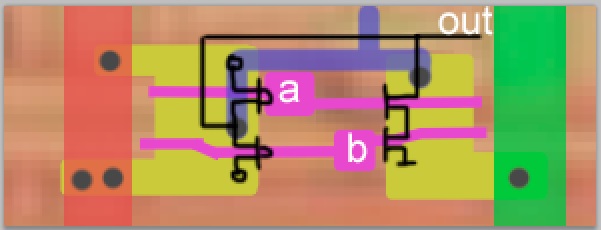
x = a & b



# NAND

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| x | output |

x = ~(a & b)

# NAND2X

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| x | output |

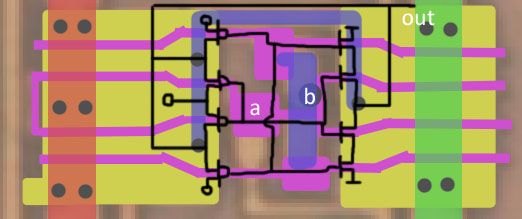
x = ~(a & b)

# NAND3X

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| x | output |

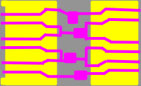
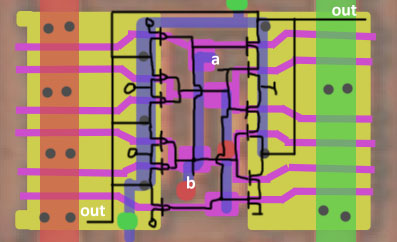
x = ~(a & b)

# NAND4X

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| x | output |

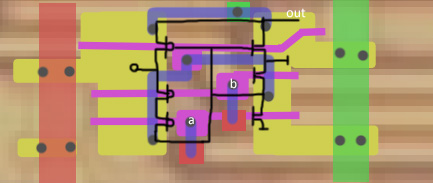
x = ~(a & b)

# OR

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| x | output |

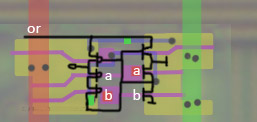
x = a | b

# OR2X

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| x | output |

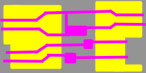
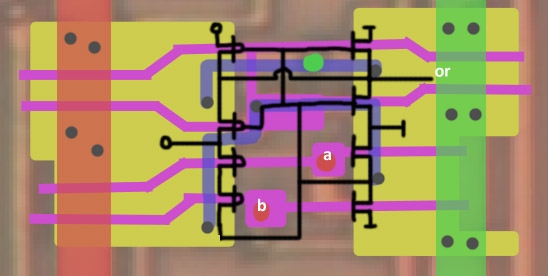
x = a | b

# OR3X

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| x | output |

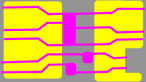
x = a | b

# OR4X

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| x | output |

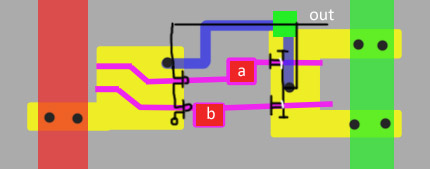
x = a | b



# NOR

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| x | output |

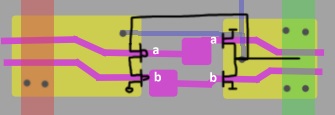
x = ~(a | b)

# NOR2X

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| x | output |

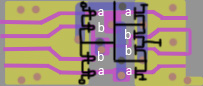
x = ~(a | b)

# NOR3X

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| x | output |

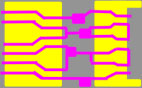
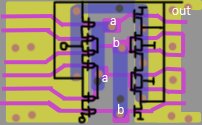
x = ~(a | b)

# NOR4X

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| x | output |

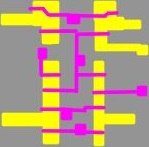
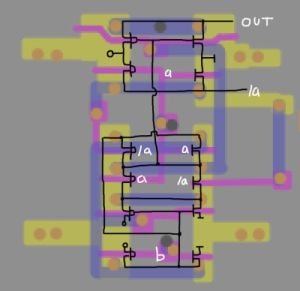
x = ~(a | b)

# XNOR1

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| x | output |

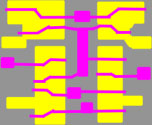
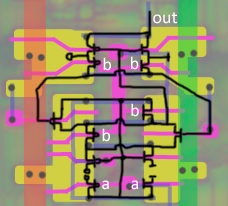
x = ~(a ^ b)

# XNOR

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| x | output |

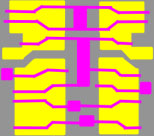
x = ~(a ^ b)

# XNOR2

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| x | output |

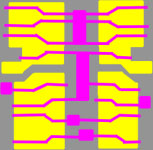
x = ~(a ^ b)



# XNOR3

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| x | output |

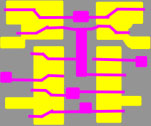
x = ~(a ^ b)



# XOR

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| x | output |

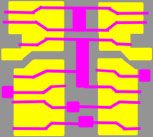
x = a ^ b



# XOR2

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| x | output |

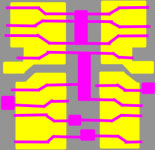
x = a ^ b



# XOR3

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| x | output |

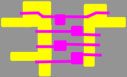
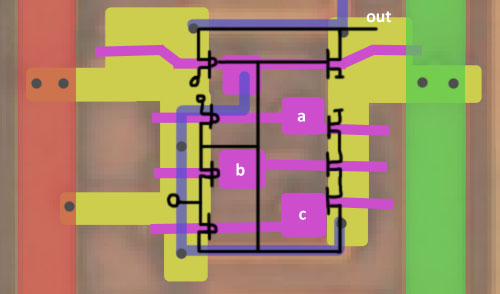
x = a ^ b



# 3-AND

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |

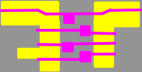
x = a & b & c

# 3-AND2X

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |

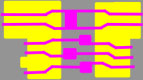
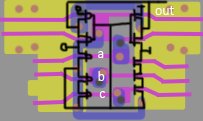
x = a & b & c



# 3-AND3X

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |

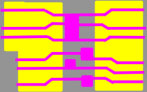
x = a & b & c

# 3-AND4X

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |

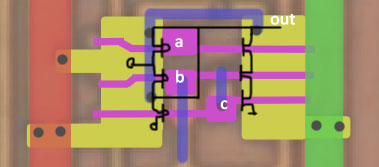
x = a & b & c



# 3-NAND

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |

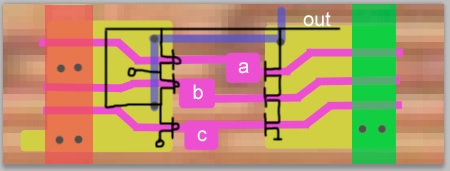
x = ~(a & b & c)

# 3-NAND2X

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |

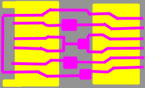
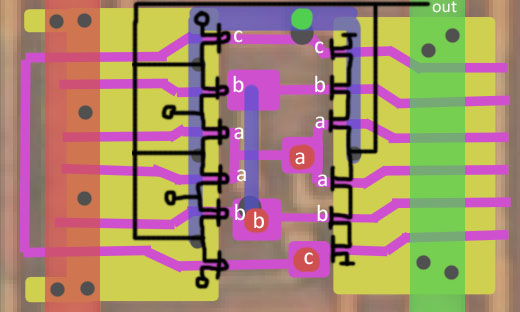
x = ~(a & b & c)

# 3-NAND4X

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |

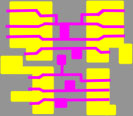
x = ~(a & b & c)

# 3-NAND5X

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |

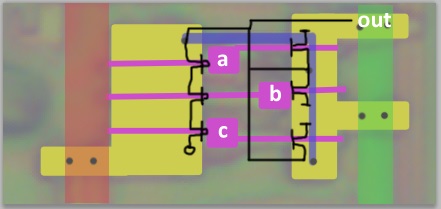
x = ~(a & b & c)



# 3-NOR

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |

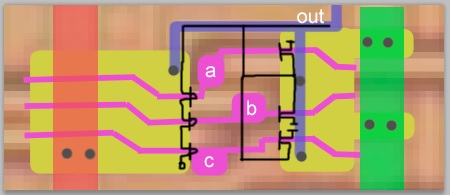
x = ~(a | b | c)

# 3-NOR2X

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |

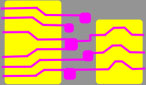
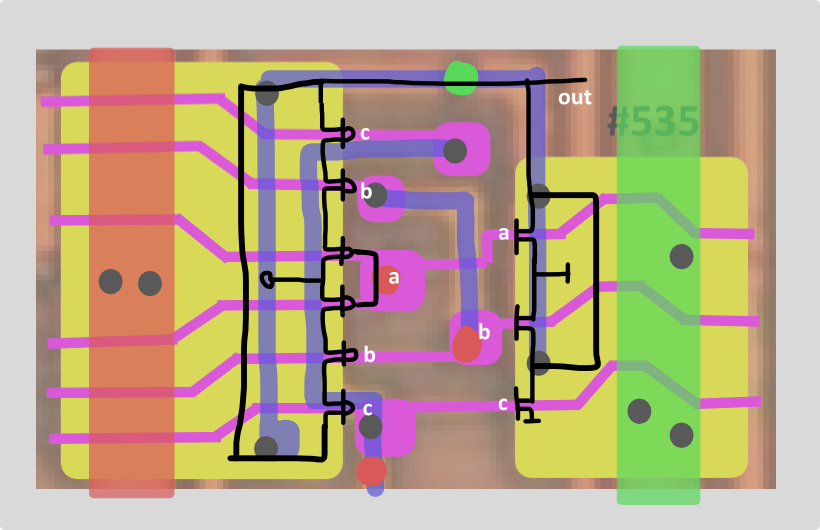
x = ~(a | b | c)

# 3-NOR4X

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |

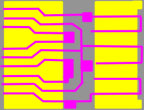
x = ~(a | b | c)

# 3-NOR6X

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |

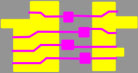
x = ~(a | b | c)



# 3-OR

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |

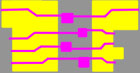
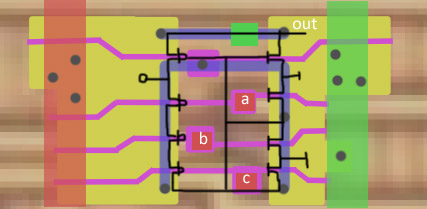
x = a | b | c

# 3-OR2X

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |

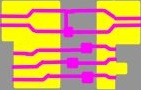
x = a | b | c

# 3-OR3X

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |

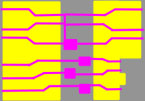
x = a | b | c



# 3-OR4X

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |

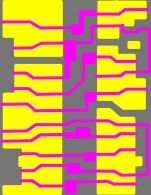
x = a | b | c



# 3-XOR

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |

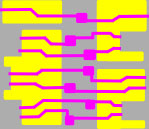
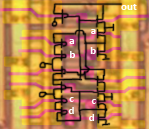
x = a ^ b ^ c



# 4-NOR

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| d | input |
| x | output |

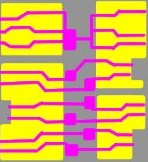
x = ~(a | b | c | d)

# 4-NOR-3X

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| d | input |
| x | output |

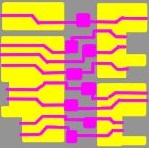
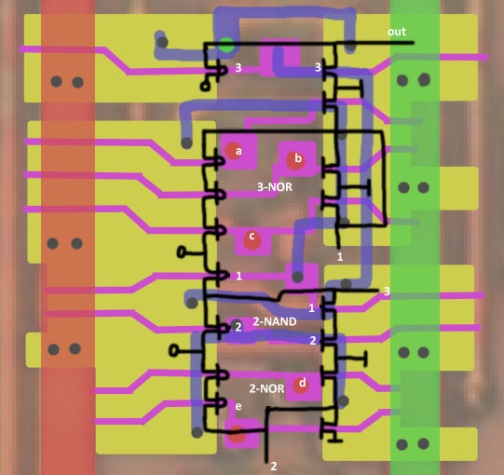
x = ~(a | b | c | d)



# 5-NOR

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| d | input |
| e | input |
| x | output |

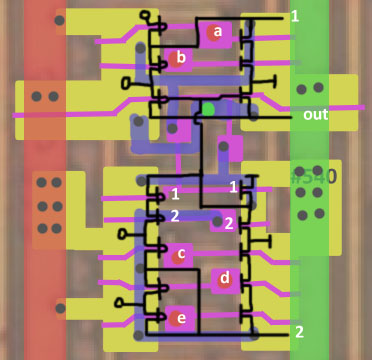
x = ~(a | b | c |d | e)

# 5-NAND

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| d | input |
| e | input |
| x | output |

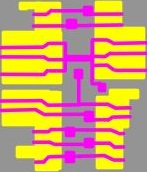
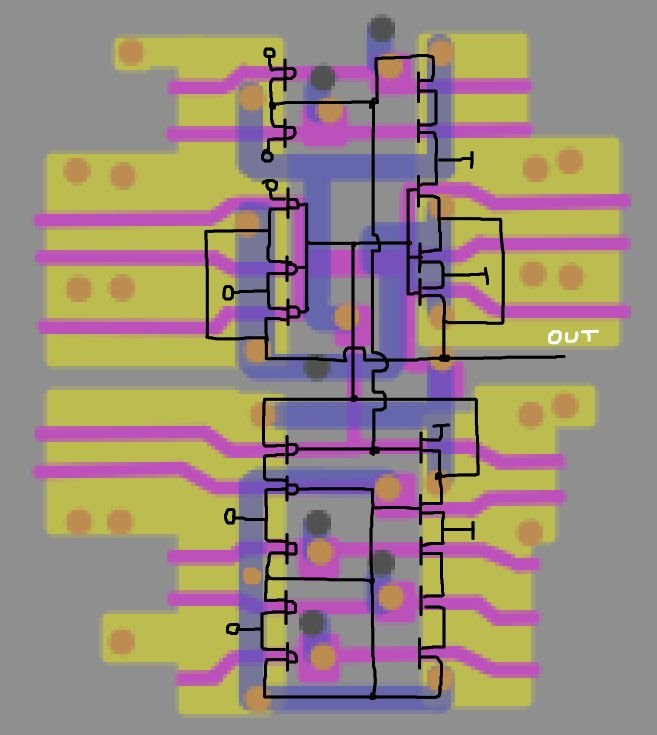
x = ~(a & b & c & d & e)

# 5-NAND3X

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| d | input |
| e | input |
| x | output |

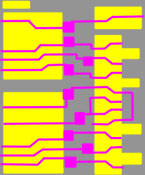
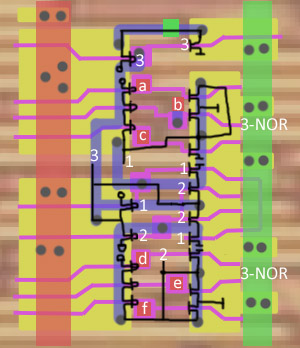
x = ~(a & b & c & d & e)

# 6-NOR

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| d | input |
| e | input |
| f | input |
| x | output |

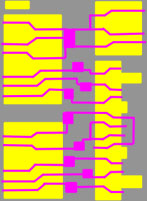
x = ~(a | b | c | d | e | f)

# 6-NOR3X

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| d | input |
| e | input |
| f | input |
| x | output |

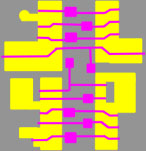
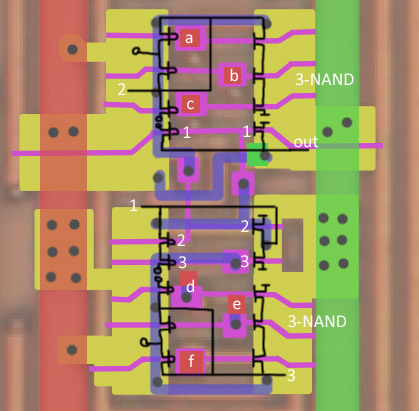
x = ~(a | b | c | d | e | f)



# 6-NAND

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| d | input |
| e | input |
| f | input |
| x | output |

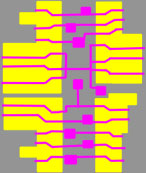
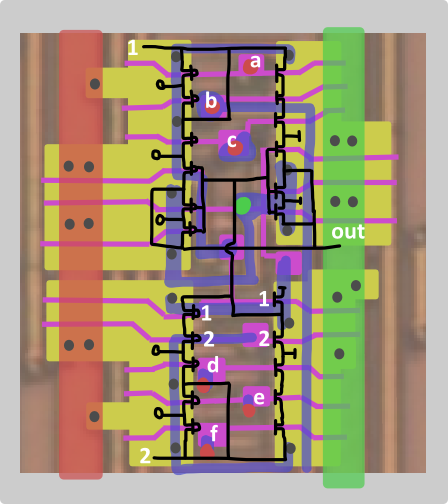
x = ~(a & b & c & d & e & f)

# 6-NAND3X

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| d | input |
| e | input |
| f | input |
| x | output |

x = ~(a & b & c & d & e & f)

# 12-AOI

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

Description.



# 12-OAI

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

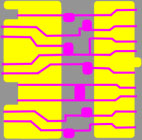
Description.



# 12-OAI3X

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

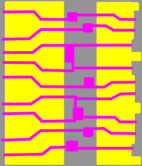
Description.



# 21-AOI3X

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

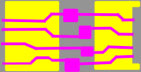
Description.



# 22-AOI

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

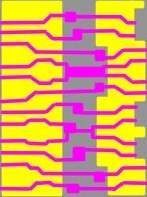
Description.



# 22-AOI3X

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

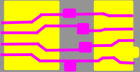
Description.



# 22-OAI

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

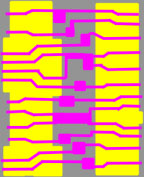
Description.



# 31-AOI

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

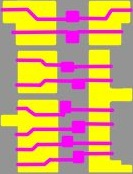
Description.



# 222-AOI

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

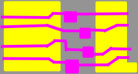
Description.



# NOR\_NAND

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

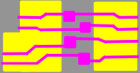
Description.



# NAND\_NOR

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

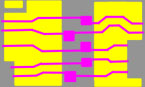
Description.



# NAND\_XOR

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

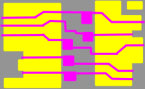
Description.



# NOR\_AND

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

Description.



# DEMUX

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

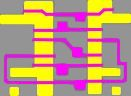
Description.



# MUX

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

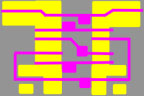
Description.



# MUX1

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

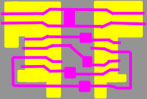
Description.



# MUX2X

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

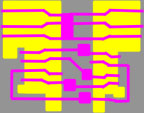
Description.



# MUX3X

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

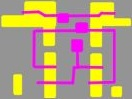
Description.



# IMUX1

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

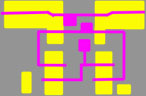
Description.



# IMUX

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

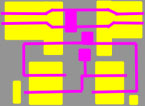
Description.



# IMUX2X

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

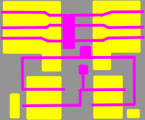
Description.



# IMUX3X

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

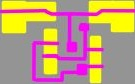
Description.



# IMUXR1

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

Description.



# IMUXR2

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

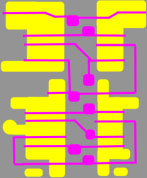
Description.



# 3-MUX

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

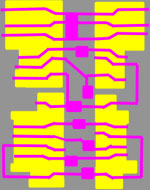
Description.



# 3-MUX3X

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

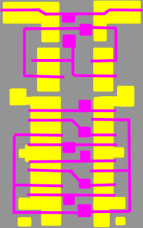
Description.



# 4-MUX

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

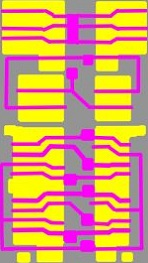
Description.



# 4-MUX3X

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

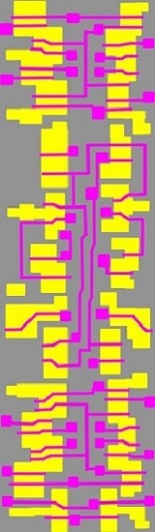
Description.



# 8-MUX

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

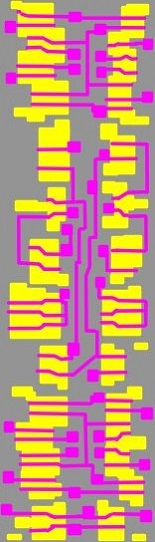
Description.



# 8-MUX3X

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

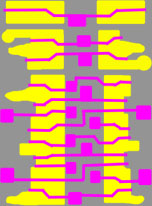
Description.



# DFF

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

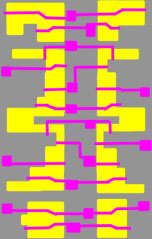
Description.



# NDFF

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

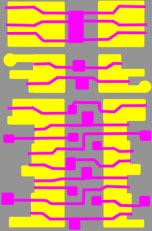
Description.



# DFF3X

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

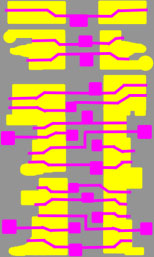
Description.



# DFFR

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

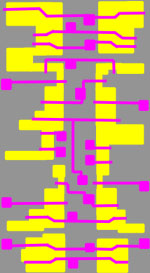
Description.



# NDFFR

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

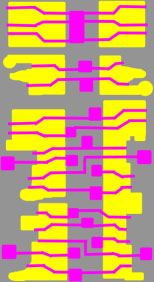
Description.



# DFFR3X

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

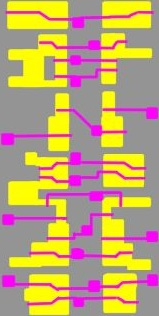
Description.



# DFFS

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

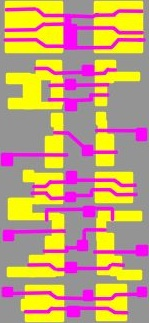
Description.



# DFFS3X

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

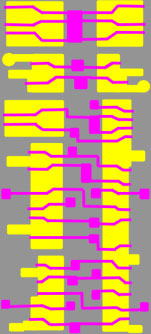
Description.



# DFFRS3X

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

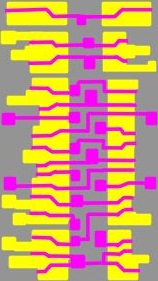
Description.



# DFF\_MUX

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

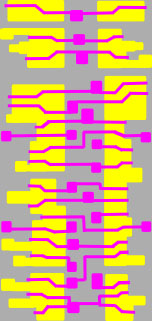
Description.



# DFFR\_MUX

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

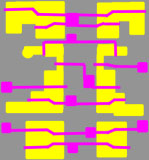
Description.



# DLATCH

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

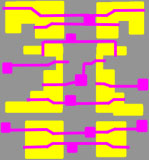
Description.



# NDLATCH

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

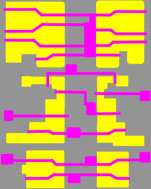
Description.



# DLATCH4X

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

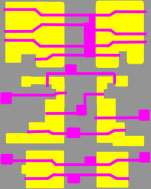
Description.



# NDLATCH4X

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

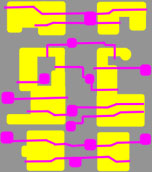
Description.



# DLATCHR2X

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

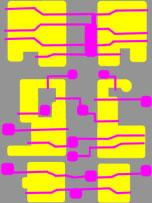
Description.



# DLATCHR4X

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

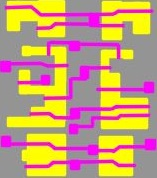
Description.



# NDLATCHR2X

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

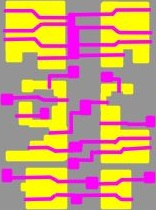
Description.



# NDLATCHR4X

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

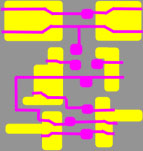
Description.



# HA

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

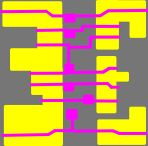
Description.



# HA2

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

Description.



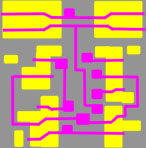
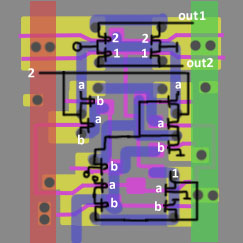
# CA

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| sum | output |
| cout | output |

This cell is similar to Full Adder in functionality, except that its Carry In always equals 1.

sum (out1) = XNOR (a,b)

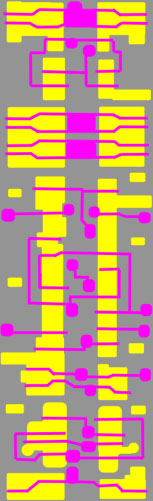
carry out (out2) = OR (a,b)

# WS1

|  |  |
| --- | --- |
| **port** | **type** |
| in1 | input |
| in2 | input |
| in3 | input |
| in4 | input |
| in5 | input |
| out1 | output |
| out2 | output |
| out3 | output |

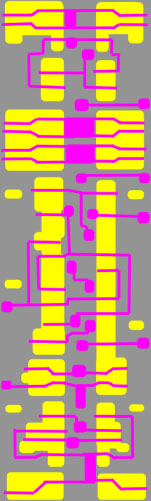
Description.



# WS2

|  |  |
| --- | --- |
| **port** | **type** |
| in1 | input |
| in2 | input |
| in3 | input |
| in4 | input |
| in5 | input |
| out1 | output |
| out2 | output |
| out3 | output |

Description.



# FA

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

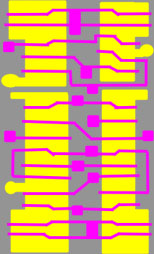
Description.



# FA2

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

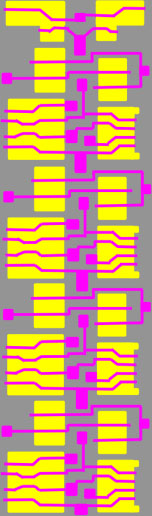
Description.



# MUX\_ARRAY

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| c1 | input |
| c2 | input |
| d1 | input |
| d2 | input |
| x | output |

Description.



# FILLER

|  |  |
| --- | --- |
| **port** | **type** |
| x0 | output |
| x1 | output |

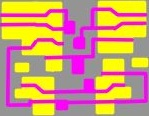
This cell was previously identified as a filler, but is most likely a 0/1 constant generator, but not connected (not used). It is present in a single instance.

# TRISTATE2

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

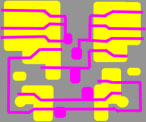
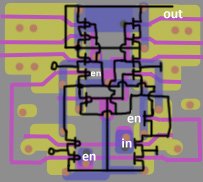
Description.



# TRISTATE3

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

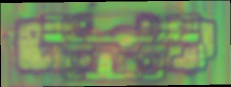
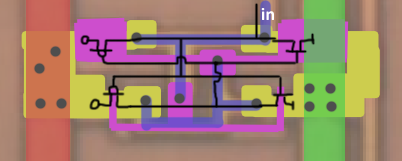
Description.

# BUS\_KEEPER

|  |  |
| --- | --- |
| **port** | **type** |
| d | input / output |

BUS keeper is to keep last state on the bus, basically, it's flip-flop constructed by two inverter, but its output drive ability is limited. Its output is connected on bus. in normal operation, it take no effect on normal logic level, but when bus is into tristate, the last logic level is kept by bus keeper to prevent bus from floating.

# CELL\_NAME

|  |  |
| --- | --- |
| **port** | **type** |
| a | input |
| b | input |
| c | input |
| x | output |
|  |  |

Description.

<picture1> <picture2>