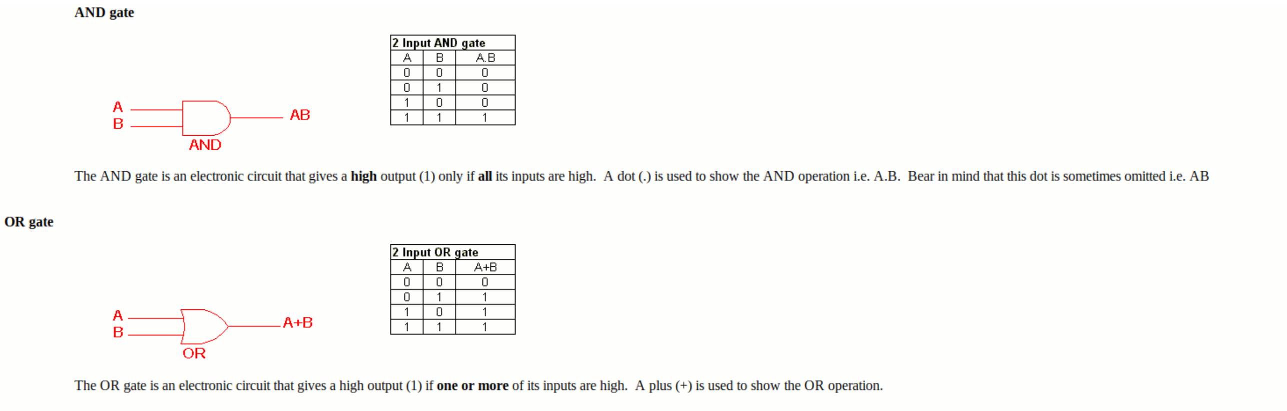
*******Draft*******

12/21/18

*******Draft*******

The FPGA is used to create hardware logic using Hardware Definition Language Verilog or VHDL. The images below were found at the following site http://www.ee.surrey.ac.uk/Projects/CAL/digital-logic/gatesfunc/#andgate.

The basic gates are AND, OR, NOT, NAND, NOR, EXOR, and EXNOR. These gates can be use to create FLIP FLOPS, and COUNTERS when connected to together. As example the two NOR gates can create a SET RESET FLIP FLOP https://www.elprocus.com/digital-electronics-flip-flop-circuit-types-and-applications/



AND gate

2 Input AND gate		
A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1

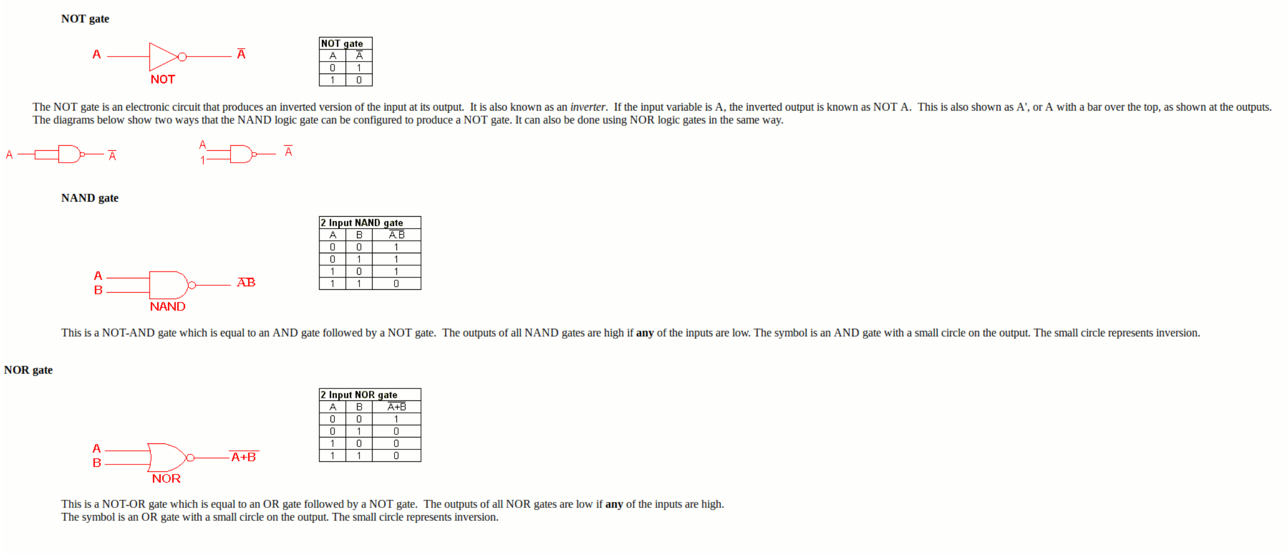
The AND gate is an electronic circuit that gives a **high** output (1) only if **all** its inputs are high. A dot (.) is used to show the AND operation i.e. A.B. Bear in mind that this dot is sometimes omitted i.e. AB

OR gate

2 Input OR gate		
A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

The OR gate is an electronic circuit that gives a high output (1) if **one or more** of its inputs are high. A plus (+) is used to show the OR operation.

NOT, NAND, NOR



NOT gate

NOT gate	
A	$\overline{A}$
0	1
1	0

The NOT gate is an electronic circuit that produces an inverted version of the input at its output. It is also known as an *inverter*. If the input variable is A, the inverted output is known as NOT A. This is also shown as A', or A with a bar over the top, as shown at the outputs. The diagrams below show two ways that the NAND logic gate can be configured to produce a NOT gate. It can also be done using NOR logic gates in the same way.

NAND gate

2 Input NAND gate		
A	B	$\overline{A.B}$
0	0	1
0	1	1
1	0	1
1	1	0

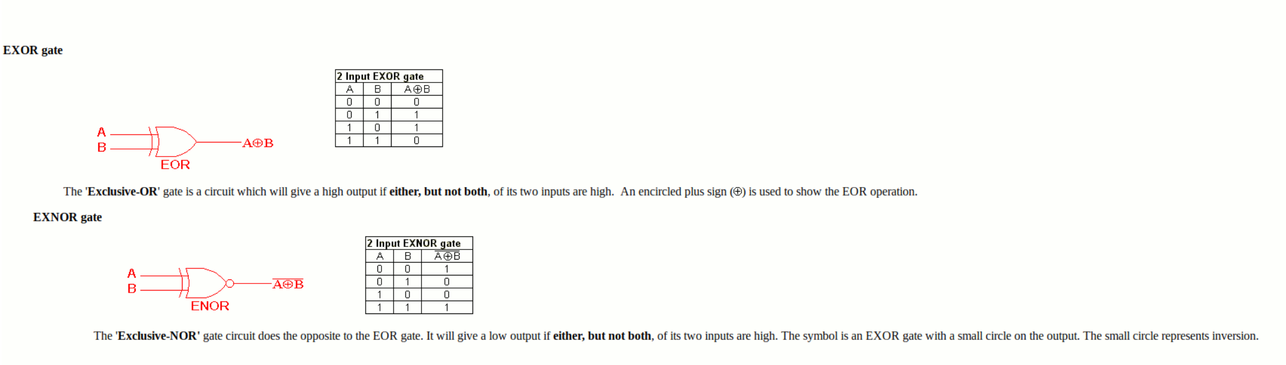
This is a NOT-AND gate which is equal to an AND gate followed by a NOT gate. The outputs of all NAND gates are high if **any** of the inputs are low. The symbol is an AND gate with a small circle on the output. The small circle represents inversion.

NOR gate

2 Input NOR gate		
A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

This is a NOT-OR gate which is equal to an OR gate followed by a NOT gate. The outputs of all NOR gates are low if **any** of the inputs are high.
The symbol is an OR gate with a small circle on the output. The small circle represents inversion.

EXOR, EXNOR



EXOR gate

2 Input EXOR gate		
A	B	A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

The '**Exclusive-OR**' gate is a circuit which will give a high output if **either, but not both**, of its two inputs are high. An encircled plus sign (⊕) is used to show the EOR operation.

EXNOR gate

2 Input EXNOR gate		
A	B	$\overline{A\oplus B}$
0	0	1
0	1	0
1	0	0
1	1	1

The '**Exclusive-NOR**' gate circuit does the opposite to the EOR gate. It will give a low output if **either, but not both**, of its two inputs are high. The symbol is an EXOR gate with a small circle on the output. The small circle represents inversion.

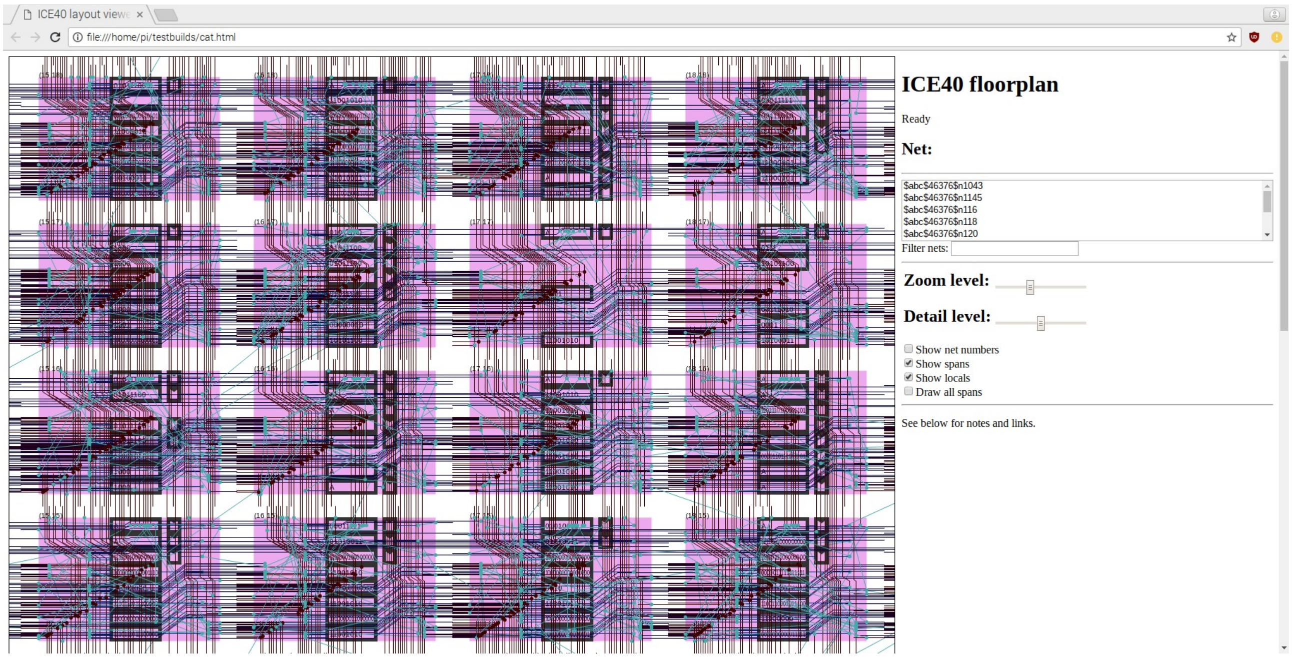
YOSYS is the program that is being used for the ice40 HX8K FPGA

In electronics,**logic synthesis** is a process by which an abstract specification of desired circuit behavior, typically at register transfer level (RTL), is turned into a design implementation in terms of logic gates, typically by a computer program called a *synthesis tool*. Common examples of this process include synthesis of designs specified in hardware description languages, including VHDL and Verilog.[1] Some synthesis tools generate bitstreams for programmable logic devices such as PALs or FPGAs, while others target the creation of ASICs. Logic synthesis is one aspect of electronic design automation.

ARACHNE-PNR is currently used this will evolve to NEXTPNR.

FPGAs, during which logic elements are placed and interconnected on the grid of the FPGA

catzip.asc This is a fairly complex design.



ICE40 floorplan

Ready

Net:

$abc$46376$n1043
$abc$46376$n1145
$abc$46376$n116
$abc$46376$n118
$abc$46376$n120

Filter nets:

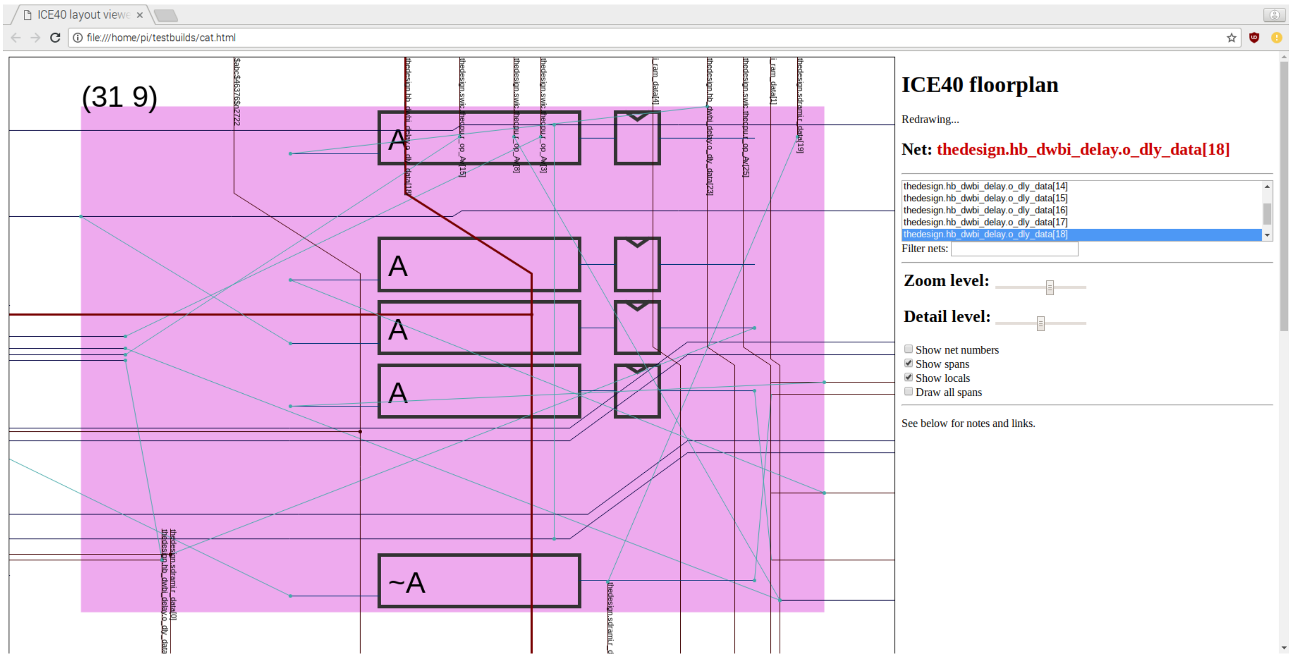
Zoom level:

Detail level:

□ Show net numbers
☑ Show spans
☑ Show locals
□ Draw all spans

See below for notes and links.

thedesign.hb_dwbi_delay.o_dly_data[18]



ICE40 floorplan

Redrawing...

Net: thedesign.hb_dwbi_delay.o_dly_data[18]

thedesign.hb_dwbi_delay.o_dly_data[14]
thedesign.hb_dwbi_delay.o_dly_data[15]
thedesign.hb_dwbi_delay.o_dly_data[16]
thedesign.hb_dwbi_delay.o_dly_data[17]
thedesign.hb_dwbi_delay.o_dly_data[18]

Filter nets:

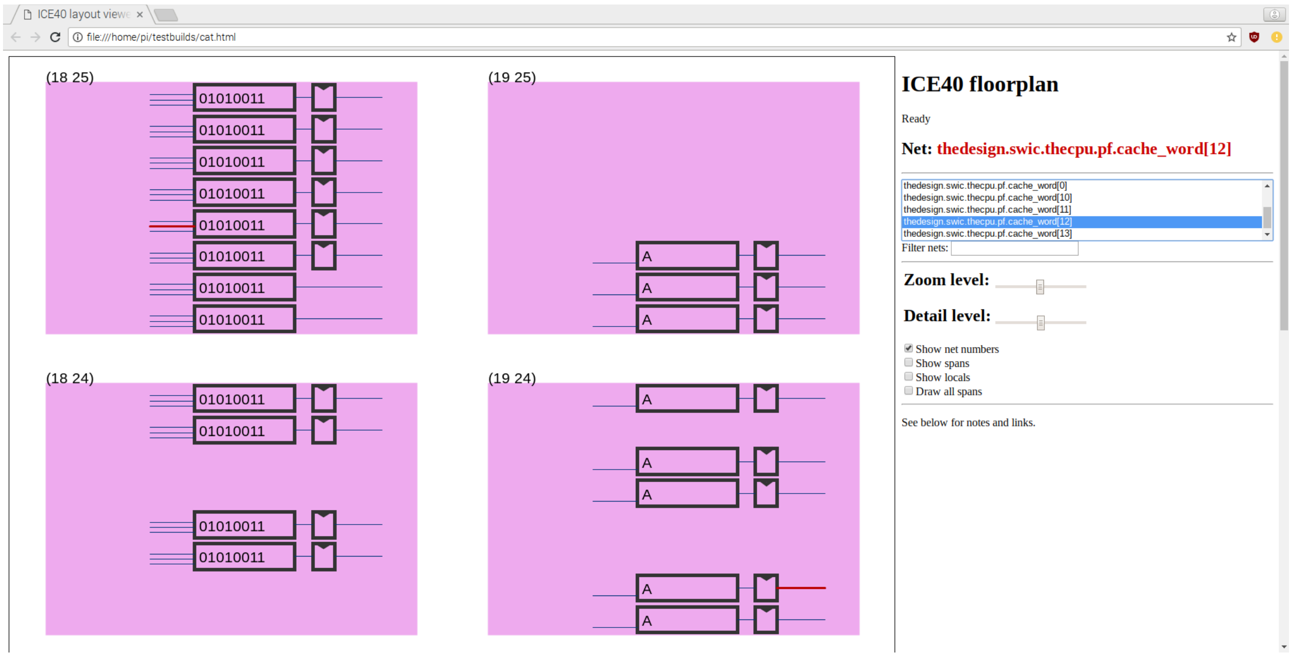
Zoom level:

Detail level:

□ Show net numbers
☑ Show spans
☑ Show locals
□ Draw all spans

See below for notes and links.

Tile



ICE40 floorplan

Ready

Net: thedesign.swic.thecpu.pf.cache_word[12]

thedesign.swic.thecpu.pf.cache_word[0]
thedesign.swic.thecpu.pf.cache_word[10]
thedesign.swic.thecpu.pf.cache_word[11]
thedesign.swic.thecpu.pf.cache_word[12]
thedesign.swic.thecpu.pf.cache_word[13]

Filter nets:

Zoom level:

Detail level:

☑ Show net numbers
□ Show spans
□ Show locals
□ Draw all spans

See below for notes and links.