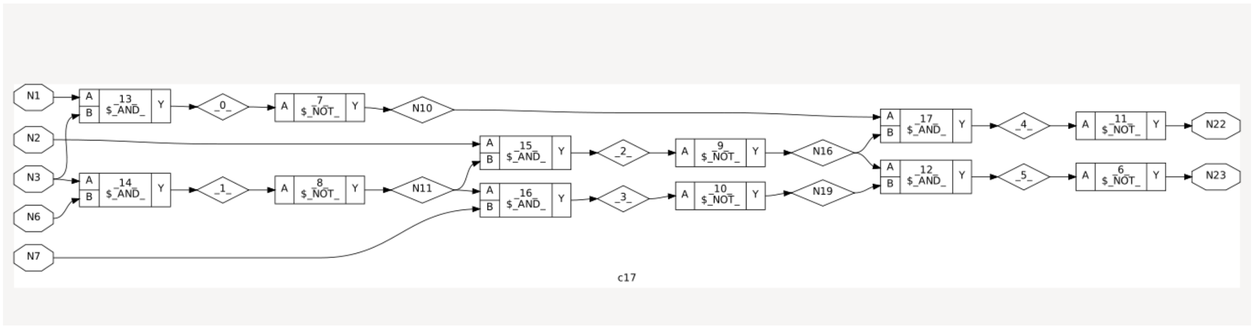
To avoid inserting INV_X1 (inverters) in your Yosys gate-level netlist, you need to understand where and why they're being inserted. They're most likely coming from this command:

abc -liberty /home/san/pro/Nangate45_typ.lib

abc maps your logic to cells in the liberty file. If INV_X1 is available in the liberty file, abc uses it during logic optimization. It adds inverters where needed to implement logic functions efficiently. Remove or comment out the INV_X1 cell from Nangate45_typ.lib, but this may break cell dependencies and cause synthesis to fail. Pass only the allowed gates to abc using the -liberty flag with a custom liberty or restrictive version. If you have another liberty file without INV_X1, use that.

If you remove this command "abc -liberty /home/san/pro/Nangate45_typ.lib" then this means your netlist will not be technology-mapped, and will stay at generic logic gate level (like $and, $or, etc.). This is useful only for simulation or logic inspection, not for physical design. It is only for RTL or debugging. : Remove abc temporarily if you're debugging or analyzing the logic .



In the synthesized gate-level graph of your c17 circuit, where is NAND2_1, which was supposed to compute

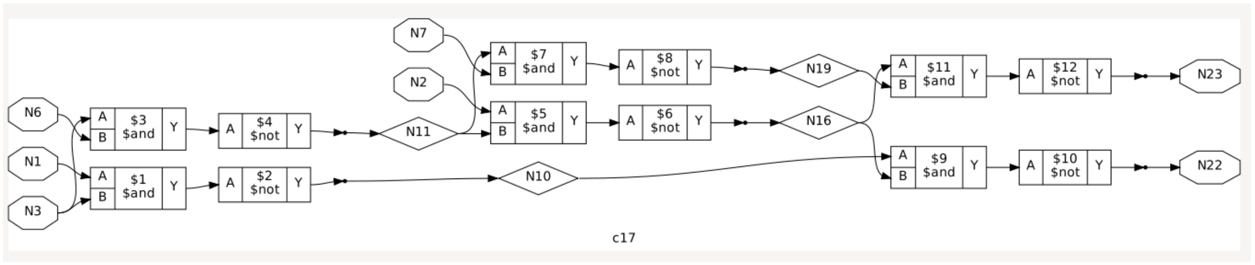
nand NAND2_1 (N10, N1, N3); NAND2_1 is optimized out by logic simplification during synthesis. It still exists logically — but its name is not preserved, and its gate may be merged or restructured. NAND2_1 is optimized out by logic simplification during synthesis. It still exists logically — but its name is not preserved, and its gate may be merged or restructured. NAND2_1 is optimized out by logic simplification during synthesis. It still exists logically — but its name is not preserved, and its gate may be merged or restructured.

read_liberty -lib -ignore_miss_dir -setattr blackbox /home/san/pro/Nangate45_typ.lib

read_verilog /home/san/pro/c17.v

hierarchy -check -c17 alu

show



Proc-

This command converts high-level procedural constructs (like always blocks, if, case, and blocking/non-blocking assignments) into lower-level RTL constructs that can be further synthesized. Translating behavioral Verilog (what you write) → structural RTL (what the tool understands and maps to gates). You must run proc if your Verilog uses:

1. always @(*) blocks
2. if, else, case, for
3. reg-typed variables
4. Blocking (=) or non-blocking (<=) assignments

Otherwise, Yosys won't know how to handle the procedural code.

proc_clean – clean up procedural code

proc_rmdead – remove unused procedural parts

proc_arst – handle asynchronous resets

proc_mux – turn if/case into mux logic

proc_dlatch, proc_dff – convert procedural registers to flops/latches

In **Yosys**, the opt command stands for **optimization**. It's a crucial step in reducing the logic complexity of your synthesized circuit and preparing it for mapping or further transformation.

In Yosys, fsm refers to a Finite State Machine optimization pass. It is used to identify, extract, and optimize FSMs in your Verilog design.

In Yosys, memory refers to a set of synthesis passes that handle memory constructs like reg [x:0] mem [y:0];, and transform them into a representation suitable for synthesis.

The techmap command replaces generic RTL cells (like $and, $or, $mux, $dff, etc.) with technology-specific gate implementations from a mapping library — typically written in Verilog.

dfflibmap -liberty /home/san/pro/Nangate45_typ.lib

This command is used to map flip-flops (registers) in your design to real flip-flop cells from your technology library (.lib file).

setundef -zero

Replaces all undefined values (x or z) in the design with concrete constants (usually 0).Hardware cannot implement unknowns (x, z) — these are simulation-only values.

Synthesis tools require all logic to resolve to 0 or 1. Ensures logic is deterministic before mapping or writing the final netlist.

clean -purge

Removes unused wires, cells, and modules (dead logic).purge ensures removal of even empty modules that are no longer instantiated.

flatten

Flattens the hierarchy of the design by inlining all submodules into the top-level module.

After this, the design becomes a single-level netlist — no instantiations of child modules remain.

Clean
The clean command removes unused wires, cells, and modules from your design. Think of it as a garbage collector for dead or redundant logic. Wires that aren't connected to anything. Cells (gates/modules) that don't affect outputs.

stat -liberty /home/san/pro/Nangate45_typ.lib

The stat command reports **statistics** about your current design: **Number of wires, cells, and gates** Area, gate count, and cell usage. Optionally uses a **Liberty (.lib)** file to show **real technology mapping stats** (e.g., area, delay)

rename -enumerate

This command **renames** all unnamed wires, cells, and other objects in your design using **unique, enumerated names**.

During synthesis and optimization steps, Yosys may generate:

- Temporary or unnamed wires like _00001_, _1_, etc.
- Auto-generated cells with generic names

These can make the design **hard to read or debug**.

write_blif

This command writes out your synthesized design in **BLIF (Berkeley Logic Interchange Format)** — a textual format used for logic synthesis tools like **ABC**, **VPR**, and **OpenFPGA**.