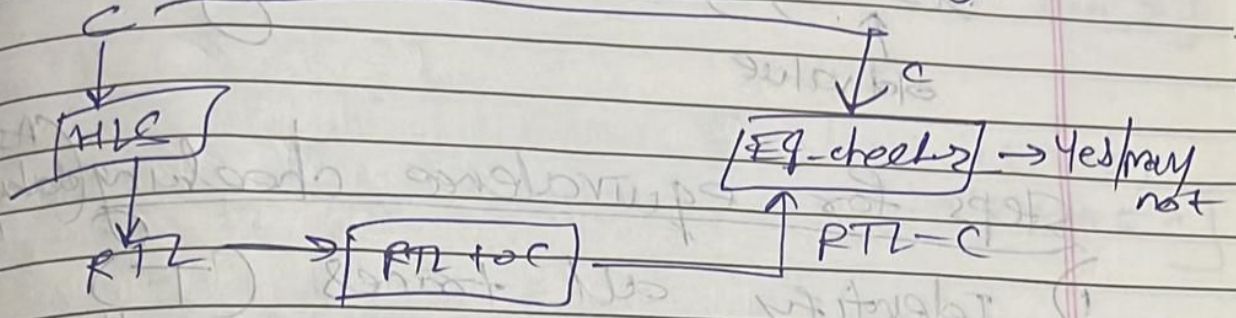
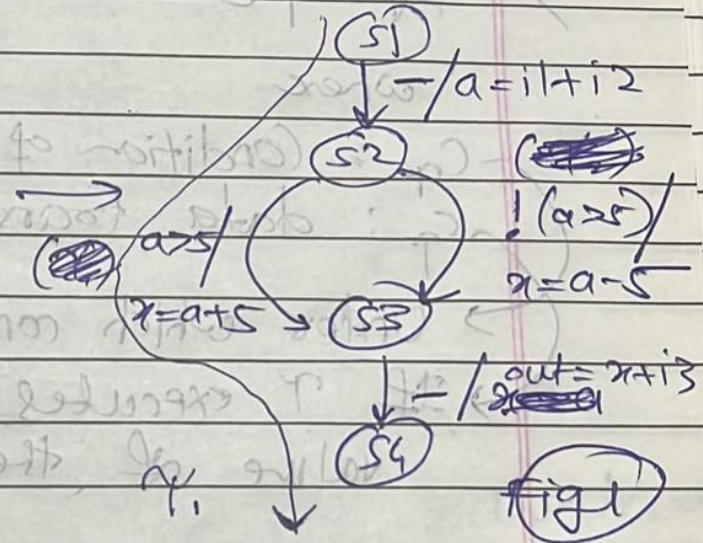
C to RTL Equivalence checking



FSM-D (Finite state machine with datapath)

```
fn (i1, i2, i3) {
    a = i1+i2;
    if (a > 5)
        x = a+5
    else
        x = a-5
    out = x+i3;
    return out;
}
```

Program → FSM-D



Note: 2 opn's in same state ⇒ opn's will be executed in parallel

Date:

* $a = b+c$
  $x = a+y$ ⟹

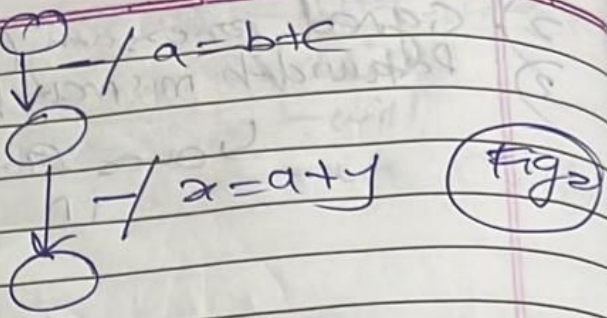
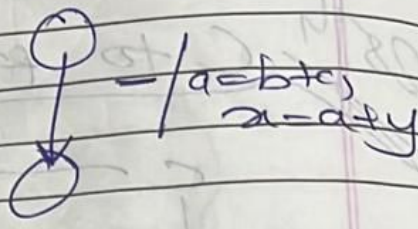


Fig 2

$a = b+c$
$x = a+y$ } parallel ⟹



↑ old value

Steps for equivalence checking (Assuming data indep, loop length 100)

1) Identify all traces (T)
→ Trace is one execution path of the program.
Eg In Fig 1 ⇒ T = 2

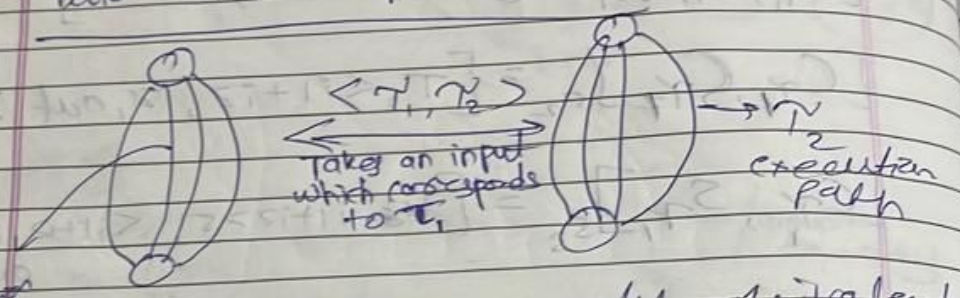
2) ∀ $\tau \in T$, C_τ & S_τ

where
C_τ : Condition of execution of τ
S_τ : data transformation of τ.

→ Under which condition τ will execute
→ If τ executes, what will be the final value of the ~~variables~~ outputs.

How to do step 4 in $O(nx)$?

Data driven approach



T_1 & T_2 are ~~only equivalent~~

Coverage Driven Testing

corresponding Traces b/w two programs using coverage driven testing

Modified algo. after running step 4) is ↓

5) Merge compatible traces in T_1

6) ~~↑~~ in T_2

7) Apply data driven approach to identify corresponding traces

8) for each corresponding trace pair, $\langle T_1, T_2 \rangle$ ($T_1 \in T_1, T_2 \in T_2$),

check $T_1 \in T_2$

↓

Binary decision program

Substitution method / Symbolic substitution

Trace → only store vars getting updating

$[C_{\tau_1}, S_{\tau_1}]_{S_1} = [T, <a, x, out>]$

$[C_{\tau_2}, S_{\tau_2}]_{S_2} = [T, <i1+i2, x, out>]$

$[C_{\tau_1'}, S_{\tau_1'}]_{S_3} = [i1+i2>5, <i1+i2, i1+i2+5, out>]$

$[C_{\tau_1}, S_{\tau_1}]_{S_4} = [i1+i2>5, <i1+i2, i1+i2+5, i1+i2+i3+5>]$

3) Perform Step 1 & 2 in P_1 & P_2 (T_1, T_2)

4) For each $\tau_1 \in T_1$, identify $\tau_2 \in T_2$ s.t.

$O(n^2 \times C)$ → Time for SMT solver

$\tau_1 \equiv \tau_2$

where $\tau_1 = \tau_2$ iff

$C_{\tau_1} = C_{\tau_2}$ and

$S_{\tau_1} = S_{\tau_2}$ → ?

Two expressions

Check $C_{\tau_1} != C_{\tau_2}$

Check SAT → SAT ⇒ not equivalent

UNSAT ⇒ Equivalent