Date / /

BuildBDD (g_9 , 1) ≡ G_9 ∵ 1 is output controlling value of AND gate

⇒ BuildBDD (g_8 , 1) ≡ R_3 ∵ 1 is output non-controlling value of OR gate

- ⇒ WitnessBDD (g_8 , 1) ≡ R_3
  - ⇒ A = BuildBDD (g_4 , 1) ≡ G_4
    - ⇒ BuildBDD (c, 1) ≡ C
    - ⇒ BuildBDD (g_1 , 1) ≡ G_1
      - ⇒ BuildBDD (b, 1) ≡ B
      - ⇒ BuildBDD (d, 1) ≡ D
      - ⇒ ConstructBDD (B ∧ D) ≡ G_1
    - ⇒ ConstructBDD (C ∧ G_1) ≡ G_4
  - ⇒ B = BuildBDD (g_7 , 0) ≡ R_2
    - ⇒ WitnessBDD (g_7 , 0) ≡ R_2
      - ⇒ A' = BuildBDD (g_6 , 0) ≡ G_6
        - ⇒ BuildBDD (g_2 , 0) ≡ R_1
          - ⇒ WitnessBDD (g_2 , 0) ≡ R_1
            - ⇒ BuildBDD (e, 0) ≡ E
            - ⇒ BuildBDD (~c, 1) ≡ ~C
            - ⇒ Restrict (E, ~C) ≡ R_1
        - ⇒ BuildBDD (d, 0) ≡ ~D
        - ⇒ ConstructBDD (R_1 ∧ ~D) ≡ G_6
      - ⇒ B' = BuildBDD (g_3 , 1) ≡ G_3
        - ⇒ BuildBDD (a, 1) ≡ A
        - ⇒ BuildBDD (b, 1) ≡ B
        - ⇒ ConstructBDD (A ∧ B) ≡ G_3
      - ⇒ RestrictBDD (G_6, G_3) ≡ R_2
  - ⇒ RestrictBDD (G_4, R_2) ≡ R_3 Have a nice time

⇒ BuildBDD (f, 1) ≡ F

⇒ ConstructBDD (R_3 ∧ F) ≡ G_9

```
BddNode a(bm.getSupport(1));
BddNode b(bm.getSupport(2));
BddNode c(bm.getSupport(3));
BddNode d(bm.getSupport(4));
BddNode e(bm.getSupport(5));
BddNode f(bm.getSupport(6));

BddNode A = a;
BddNode B = b;
BddNode C = c;
BddNode D = d;
BddNode E = ~e;
BddNode F = f;
BddNode C_bar = ~C;
BddNode D_bar = ~D;

BddNode G1 = B & D;
BddNode G4 = C & G1;
BddNode R1 = bm.restrict(E, C_bar);
BddNode G6 = R1 | D_bar;
BddNode G3 = A & B;
BddNode R2 = bm.restrict(G6, G3);
BddNode R3 = bm.restrict(G4, R2);
BddNode G9 = R3 & F;

cout << G9 << endl;
```

```
[6](+) 0x117dc90 (1)
  [4](+) 0x117dba0 (3)
    [3](+) 0x117db50 (1)
      [2](+) 0x117d970 (5)
        [0](+) 0x117c310 (19)
        [0](-) 0x117c310 (19) (*)
      [0](-) 0x117c310 (19) (*)
    [0](-) 0x117c310 (19) (*)
  [0](-) 0x117c310 (19) (*)

==> Total #BddNodes : 5

```

difference: 5