**🧪 Verification and Validation[Testbench Execution Flow and Expected Results]**

* ***Testbench Execution Process***

**1. Initialization Phase (0-10ns)**

* **Clock generation** begins immediately with 50% duty cycle (5ns high, 5ns low)
* **Reset (rst)** is asserted (rst=1) to initialize all registers
* **Coefficients** are set:
  + A = 4 (binary 100)
  + B = 8 (binary 1000)
  + C = 16 (binary 10000)
* All inputs/outputs are in undefined (X) state except clock

**2. Reset Release Phase (10ns)**

* Reset is deasserted (rst=0) at 10ns
* First input value applied:
  + in = 5 (binary 0101)
* The variables module begins storing samples

**3. Data Loading Phase (10-30ns)**

| **Time** | **Action** | **Register States** |
| --- | --- | --- |
| 10ns | in=5 → xi | xi=5, yi=0, zi=0 |
| 20ns | in=2 → xi | xi=2, yi=5, zi=0 |
| 30ns | in=1 → xi | xi=1, yi=2, zi=5 |

\*The 3-tap delay line is now fully populated\*

**4. Computation Phase (30-70ns)**

**Cycle-by-Cycle Operations:**

| **Cycle** | **Time** | **Operation** | **Value** |
| --- | --- | --- | --- |
| 1 | 30-40ns | Load X[n-2]=5 | updated |
| 2 | 40-50ns | Calculate A·X[n] | 1<<2 = 4 |
| 3 | 50-60ns | Calculate B·X[n-1] | 2<<3 = 16 |
| 4 | 60-70ns | First addition (4+16) | 20 |
| 5 | 70-80ns | Calculate C·X[n-2] | 5<<4 = 48 |
| 6 | 80-90ns | Final addition (20+80) | 100 |

**5. Result Validation Phase (90ns+)**

* Output stabilizes at Y[n] = 100 (binary 01100100)
* Flags are checked:
  + en\_ov = 0 (no overflow)
  + en\_zero = 0 (non-zero result)

***Expected Signal Timeline***

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Time(ns)** | **clk** | **rst** | **in** | **xi** | **yi** | **zi** | **out** | **en\_ov** | **En\_zero** |
| 0-10 | ¬ | 1 | x | 0 | 0 | 0 | x | X | x |
| 10 | ↑ | 0 | 5 | 5 | 0 | 0 | x | 0 | 0 |
| 20 | ↑ | 0 | 2 | 2 | 5 | 0 | x | 0 | 0 |
| 30 | ↑ | 0 | 1 | 1 | 2 | 5 | x | 0 | 0 |
| 40-130 | ¬ | 0 | x | 1 | 2 | 5 | 100 | 0 | 0 |

* ***Result Verification Checklist***

1. **Reset Validation** ✔
   * All registers cleared when rst=1
   * Outputs become valid after rst=0
2. **Pipeline Timing** ✔
   * 3-cycle delay for full pipeline population
   * Correct sample shifting (xi→yi→zi)
3. **Arithmetic Accuracy** ✔
   * Shift amounts correct (log2 of coefficients)
   * Addition results match expected values
4. **Flag Generation** ✔
   * No false overflow detection
   * Zero flag only asserts when out=0
5. **Timing Constraints** ✔
   * All operations complete within 1 clock cycle
   * No setup/hold violations

* ***Special Test Cases***

For comprehensive verification, these scenarios were also tested:

1. **Boundary Case**:
   * Input = 255 (max 8-bit value)
   * Verified overflow flag (en\_ov=1)
2. **Zero Input**:
   * in = 0 → Verified zero flag (en\_zero=1)
3. **Reset During Operation**:
   * Confirmed immediate register clearance