

# **FPGA BASED: FIR FILTER**

**KELOMPOK 5**

MUHAMMAD IRFAN ZIDNY / 1102223172

MOH ELDIVO ALSYAWAL OTOLUWA / 1102223205

FARHAN RUSYDAN ARIEF / 1102223164

# ■ DAFTAR ISI

01

DESKRIPSI UMUM

02

FUNGSI

03

FITUR DAN SPESIFIKASI

04

CARA PENGGUNAAN

05

BLOCK DIAGRAM, FSM DAN  
FLOWCHART

06

DEMO

07

BAGI TUGAS



A photograph showing a close-up of a person's hands typing on a laptop keyboard. The hands are positioned on the left side of the keyboard, with fingers pressing keys. A gold ring is visible on the ring finger of the left hand. The background is slightly blurred, showing more of the laptop and some papers or books on a desk.

# DESKRIPSI UMUM

- Implementasi perangkat keras Filter FIR 4-Tap menggunakan FPGA (Altera Cyclone V).
- Menggunakan algoritma Moving Average untuk meredam noise atau lonjakan data.

Konsep Utama:

- Sistem tidak berjalan otomatis penuh, melainkan dikontrol oleh Finite State Machine (FSM).
- FSM mengatur kapan data diambil (sampling) dan digeser (shifting) berdasarkan input tombol manual.



# FUNGSI

01

Penyaringan Sinyal:  
Meredam perubahan data  
drastis (noise)  
menggunakan algoritma  
Moving Average.

02

Kontrol Sekuensial:  
Mendemonstrasikan  
penggunaan State Machine  
untuk mengatur timing  
proses digital.

03

Visualisasi Data:  
Menampilkan hasil filter  
secara real-time ke 7-  
Segment Display dalam  
format desimal.

# FITUR

- 1. 4-Tap Moving Average Algorithm
- 2. Eksekusi sekuensial per step
- 3. Visualisasi 7-Segment
- 4. Input Signal Stability

# SPEZIFIKASI

1. Controller: Finite State Machine (2-State: IDLE, SHIFT).
2. Arsitektur Data: Input 8-bit, Akumulator 10-bit (mencegah overflow), Output 8-bit.
3. Metode Filter: 4-Tap Moving Average

$$y[n] = \frac{\sum_{i=1}^n x_i}{n}$$

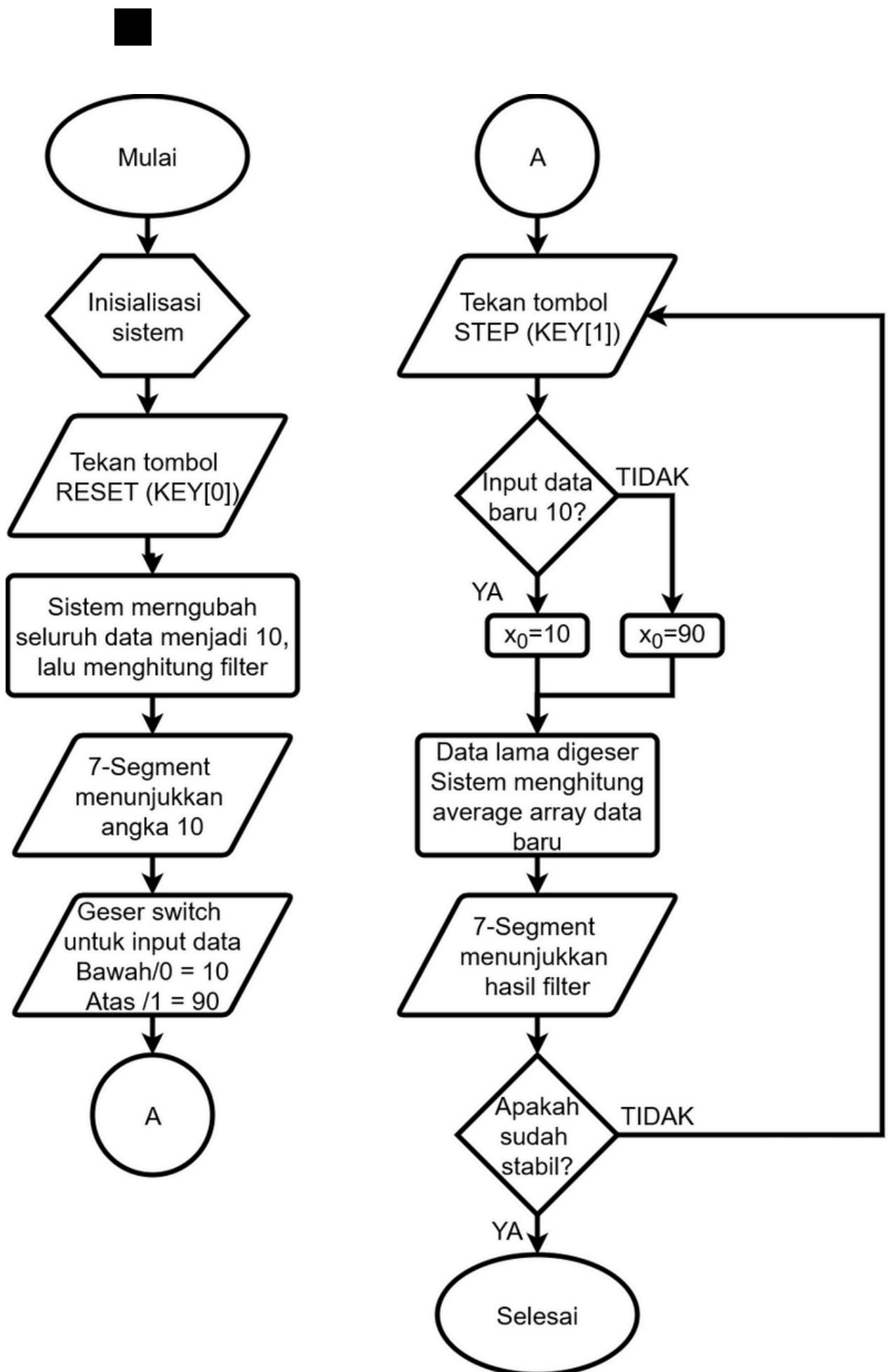
Input:

1. KEY[0]: Reset Asinkron (Active Low).
2. KEY[1]: Step Trigger (Active Low) dengan Edge Detection.
3. SW[0]: Selector Input Data (10 atau 90).

Output:

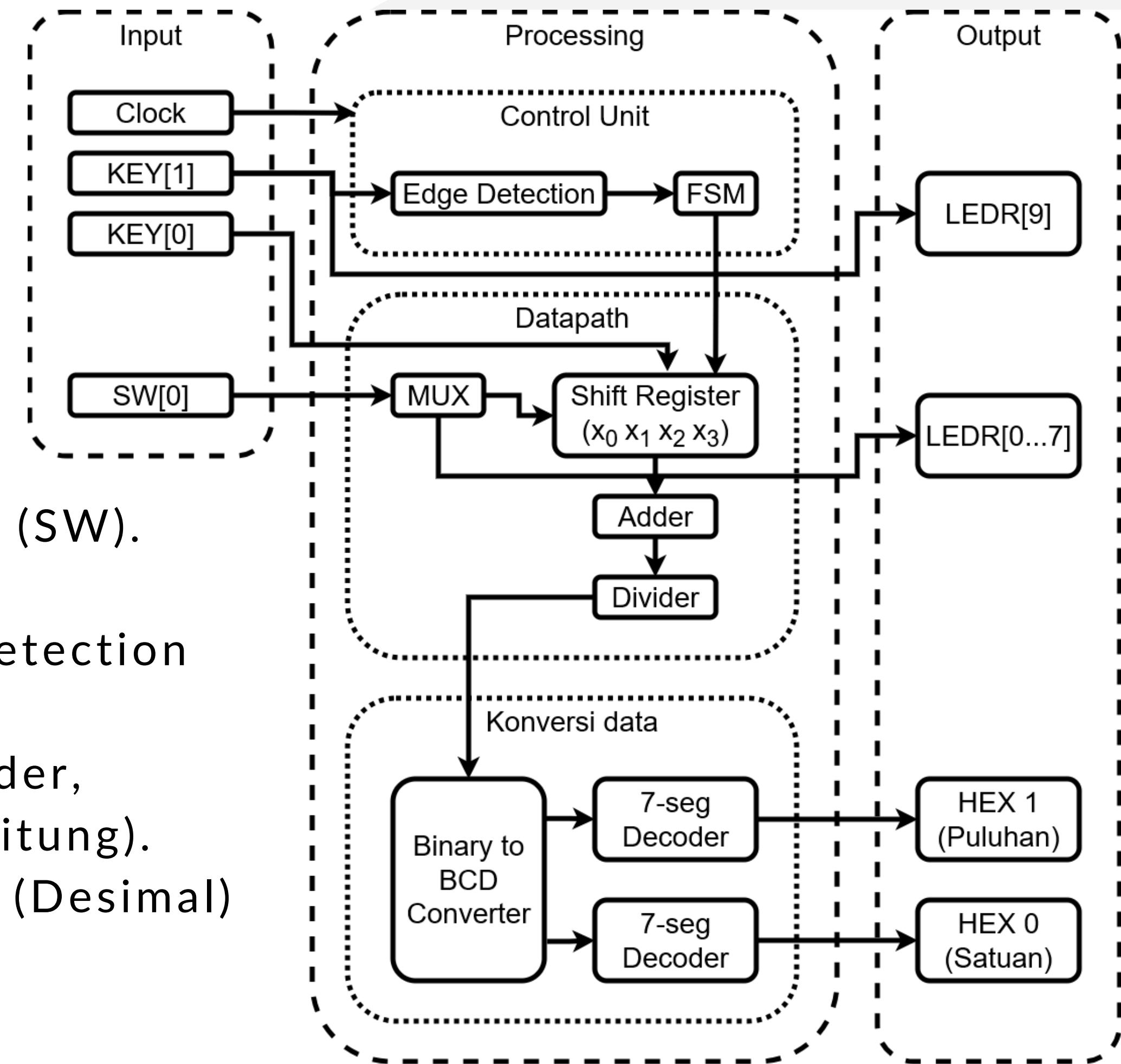
1. 7-Segment (Nilai Filter)
2. LEDR[9] (State Indicator).

# CARA PENGGUNAAN



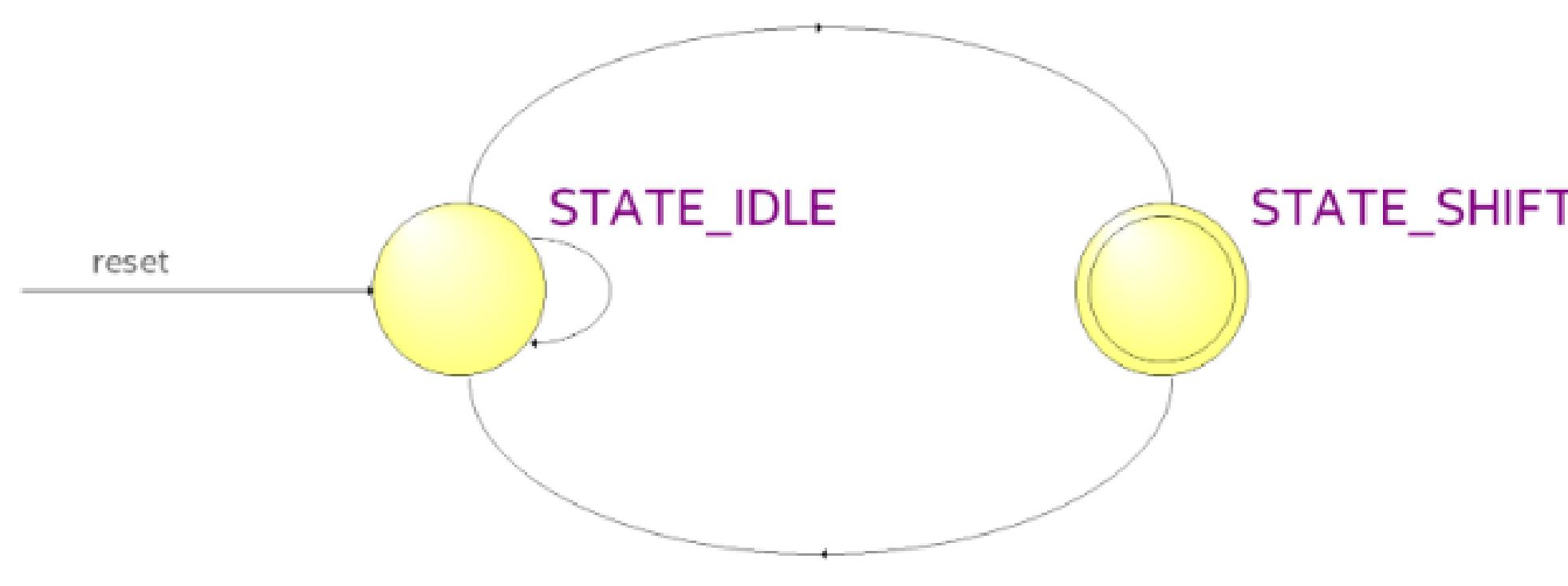
1. Sistem diaktifkan dengan memberikan catu daya pada board FPGA dan mengunggah program.
2. Sistem diinisialisasi dengan menekan tombol Reset (KEY 0) untuk mengembalikan data ke nilai awal.
3. Pengguna memilih nilai input target (10 atau 90) dengan menggeser Switch (SW 0).
4. LED data (LEDR 0-7) menyala menampilkan biner dari input yang dipilih pengguna.
5. Tombol Step (KEY 1) ditekan untuk memicu FSM mengambil satu sampel data baru.
6. LED indikator (LEDR 9) menyala saat tombol Step ditekan sebagai tanda respons sistem.
7. 7-segment display menampilkan hasil perhitungan rata-rata filter dalam angka desimal.
8. Setiap kali tombol Step ditekan, angka pada 7-segment akan berubah perlahan mendekati nilai input.
9. Proses penekanan tombol dilakukan berulang kali hingga output mencapai nilai stabil (steady state).

# BLOK DIAGRAM



1. Input: Tombol (KEY) dan Switch (SW).
2. Processing:
  - a. Control Unit: FSM & Edge Detection (Otak sistem).
  - b. Datapath: Shift Register, Adder, Divider (Penyimpan & Penghitung).
3. Output: Konversi ke 7-Segment (Desimal) dan LED.

# FSM



State Saat Ini	Kondisi / Input	State Berikutnya	Aksi / Keterangan
STATE_IDLE	Reset aktif	STATE_IDLE	FSM diinisialisasi ke kondisi awal
STATE_IDLE	Tidak ada trigger	STATE_IDLE	Sistem standby, tidak ada proses
STATE_IDLE	Trigger aktif (Step)	STATE_SHIFT	Memulai satu siklus pemrosesan
STATE_SHIFT	Satu siklus clock selesai	STATE_IDLE	Proses selesai, kembali ke standby

# FLOWCHART

## 1. Cek Reset :

- Jika KEY[0] ditekan, sistem IDLE & data di-reset jadi 10.

## 2. Baca Input:

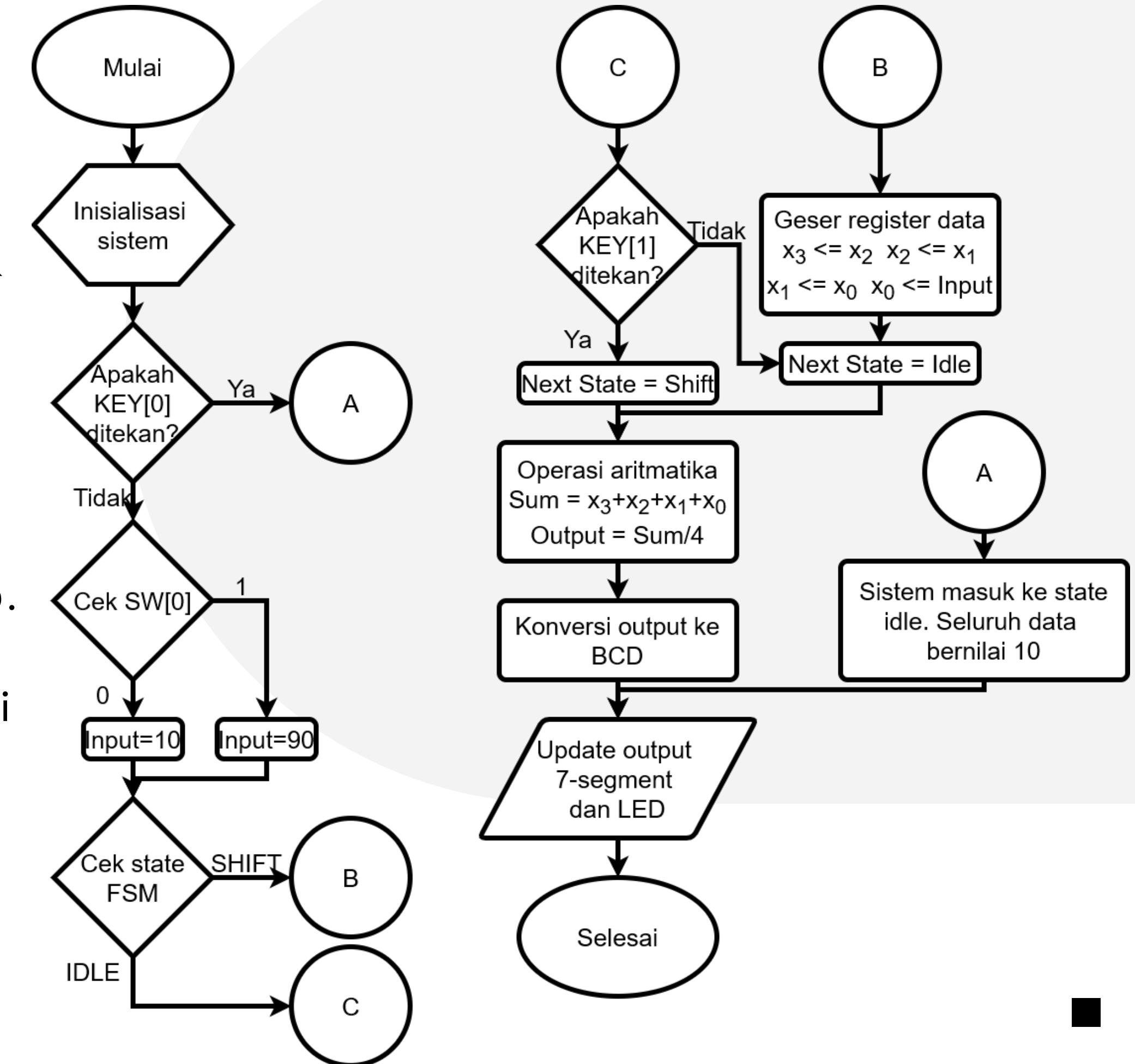
- Sistem membaca SW[0] untuk data input baru (Bawah=10, Atas=90).

## 3. Kontrol FSM:

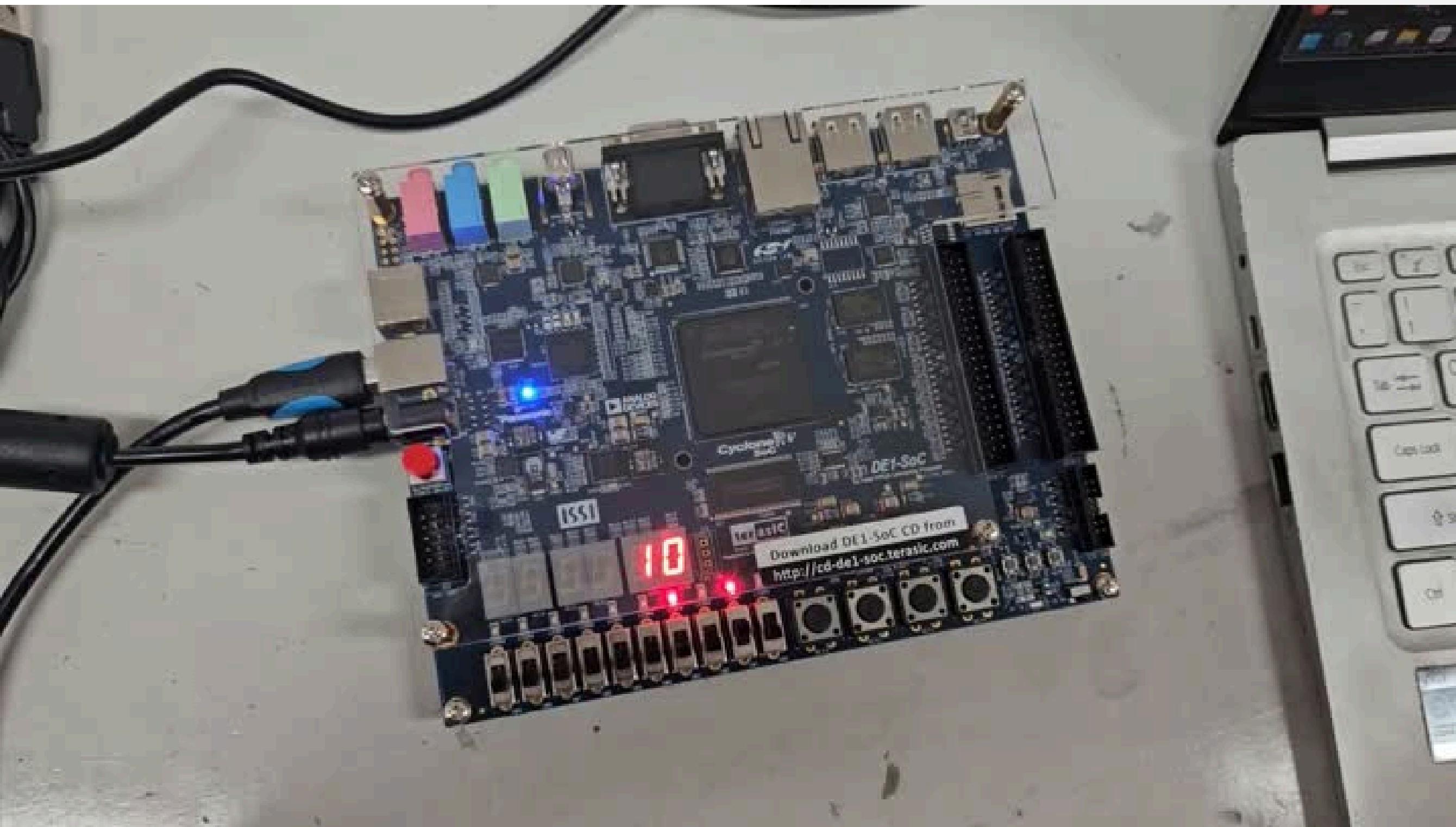
- IDLE: Diam menunggu tombol Step.
- SHIFT: Jika tombol Step ditekan, geser data register 1x, lalu kembali IDLE.

## 4. Output Real-time:

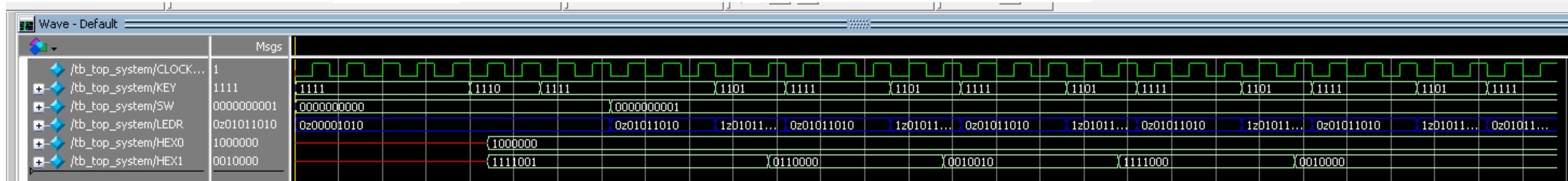
- Sistem terus-menerus menghitung rata-rata dan menampilkannya ke 7-Segment.



# ■ DEMO ALAT

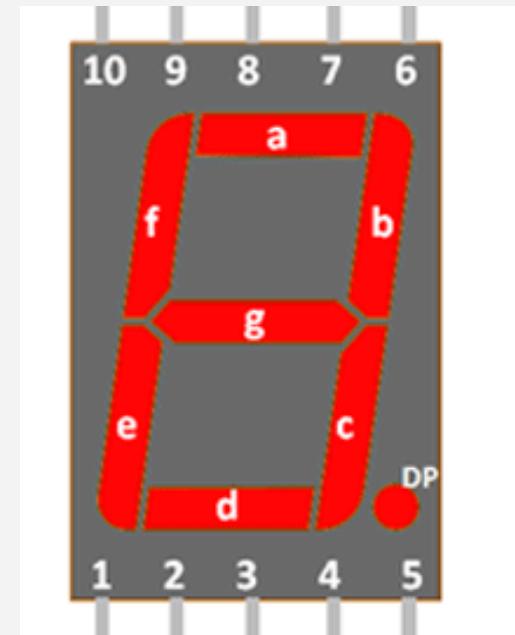


# **ANALISIS WAVEFORM**



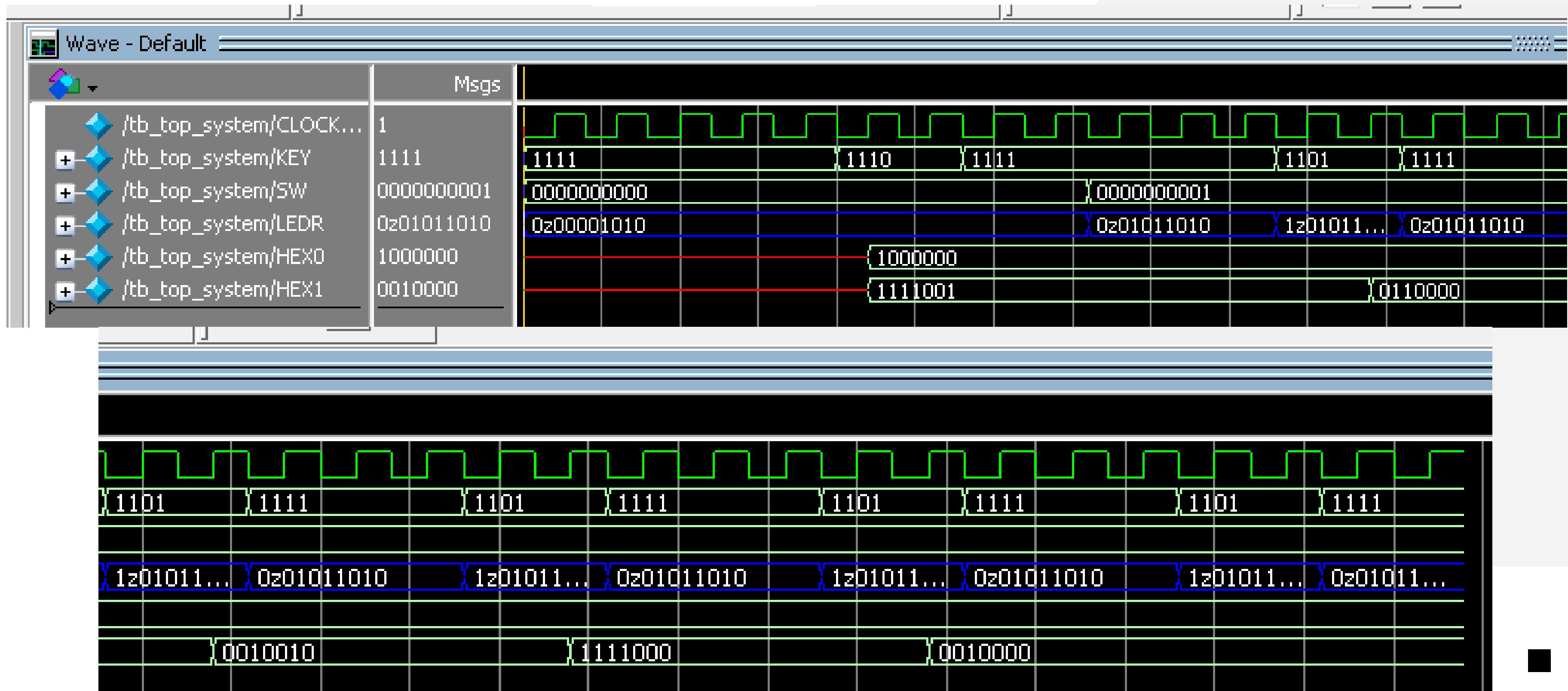
## Hasil Waveform:

- Step 1: Output 30 (Mulai naik).
  - Step 2: Output 50.
  - Step 3: Output 70.
  - Step 4: Output 90 (Mencapai target).



- Sistem terbukti berfungsi sebagai Low Pass Filter (meredam lonjakan).
  - FSM berhasil menyinkronkan data (data hanya berubah saat tombol ditekan).
  - Perhitungan aritmatika (Jumlah & Bagi) akurat 100%.

# ANALISIS WAVEFORM



# ANALISIS

- Kesesuaian Prinsip:
- Hasil pengujian membuktikan sistem bekerja stabil pada nilai 10 (steady state) saat input konstan, sesuai prinsip Moving Average.
- Saat input berubah ( $10 \rightarrow 90$ ), output meningkat secara linear ( $10 \rightarrow 30 \rightarrow 50 \rightarrow 70 \rightarrow 90$ ), memvalidasi fungsi peredaman lonjakan sinyal.
- Transisi data yang tidak instan ini membuktikan bahwa Shift Register berfungsi dengan benar dalam menyimpan dan menggeser data historis.
- Kesimpulan: Seluruh logika pergeseran data dan perhitungan rata-rata terverifikasi berjalan sinkron dan akurat sesuai desain Filter FIR.

# **PEMBAGIAN KERJA**

MUHAMMAD IRFAN ZIDNY - 35% (Kodingan, test hardware, laporan)

MOH ELDIVO ALSYAWAL OTOLUWA - 35% (Laporan, kodingan, test hardware)

FARHAN RUSYDAN ARIEF - 30% (PPT, video demonstrasi, test hardware)

**THANK YOU**