

FPGA-Based Multi-Channel Digital Sampling Oscilloscope

Maxwell Onyango

Department of Electrical Engineering
and Computer Science
Massachusetts Institute of Technology
maxmlewa@mit.edu

Mensur Muzyin

Department of Electrical Engineering
and Computer Science
Massachusetts Institute of Technology
mensurm@mit.edu

Shafick Kayemba

Department of Electrical Engineering
and Computer Science
Massachusetts Institute of Technology
kshafick@mit.edu

Abstract—This paper presents the design of a digital sampling oscilloscope (DSO) implemented on the RealDigital Urbana development board with an auxiliary analog front-end for signal acquisition and waveform generator output. The system is designed to integrate two 12-bit dual channel data acquisition pipelines, making a total of 4 channels. It also hosts a direct digital synthesis (DDS) based waveform generator, and a real-time display system that renders captured waveforms on an external monitor via HDMI. The major functional blocks (data acquisition, storage, measurements, and display) are implemented entirely on FPGA hardware. The analog front-end maps ± 5 V laboratory signals into the PMOD AD1 ADC's 0–3.3 V range using level shifting and filtering, enabling both DC and AC measurements. The final prototype achieves 1 MSPS dual-channel sampling, real-time rendering at 60 Hz HDMI output, and automated gain measurement for Bode plotting over a 10 Hz–6 kHz sweep. The design targets a pedagogical, open-hardware oscilloscope platform for exploring mixed-signal, FPGA, and digital systems concepts.

Index Terms—Digital storage oscilloscope, field-programmable gate array, data acquisition, direct digital synthesis, HDMI display, waveform generation, BRAM architecture

I. INTRODUCTION

This project aims to develop a pedagogically useful digital sampling oscilloscope (DSO) implemented entirely on Field-Programmable Gate Array (FPGA) board.

Our design employs PMOD peripheral modules for analog-to-digital and digital-to-analog conversion, with high-definition multimedia interface (HDMI) serving as the display output interface onto an external monitor. All major oscilloscope subsystems, including data acquisition, measurement and scaling, waveform storage and triggering, display rendering, user control, and an integrated function generator, are implemented directly in FPGA fabric. High-definition visualization is provided via a 720p HDMI output to an external monitor, while a menu-driven interface mapped onto limited board inputs (pushbuttons, switches, and a rotary encoder) controls the instrument.

The three primary objectives that guide the project development are:

- (i) Achieving reliable live signal visualization for frequencies up to 100 kHz utilizing the Urbana development board with PMOD interface modules.

- (ii) Implementing a configurable waveform generator capable of producing standard test signals (sinusoidal, square, ramp, triangular, and DC) across a frequency range from 10Hz to 100 kHz.
- (iii) Developing automated Bode plot generation for frequency response analysis through integration of the waveform generation and measurements subsystems.

This report presents on the current integrated prototype, which implements a dual-channel, 12-bit data acquisition pipeline, a DDS-based function generator, BRAM-based waveform storage with triggering, a measurements and scaling block, and a 720p HDMI display system with cursors and legend. We describe the system architecture and module-level implementation, analyze FPGA resource utilization and timing, and discuss design tradeoffs and limitations. We then outline the path to the full four-channel, math, XY, and Bode features the final system, and reflect on the pedagogical value of a digital sampling oscilloscope built entirely on FPGA hardware.

II. OVERALL SYSTEM ARCHITECTURE

The overall system is composed of five primary functional blocks: the Data Acquisition Block, the Measurements and Scaling Block, the Bode Block, the Store Block, and the Display Block. Each block is discussed in detail below.

A. Data Acquisition

The data-acquisition (DAQ) subsystem consists of the analog front end, a pair of PMOD AD1 analog-to-digital converters, and a sampler module implemented on the FPGA.

The PMOD AD1 package integrates two AD7476A 12-bit ADCs, providing simultaneous dual-channel input [1]. The channels share a three-wire SPI-like communication protocol. We derive the ADC serial clock from the system clock and configure `data_clk` to toggle every six system-clock cycles. Each sample conversion requires 16 `data_clk` cycles followed by a four-cycle idle period, so the total number of system-clock cycles per conversion is

$$16 \times 6 + 4 = 100.$$



For our 100 MHz system clock this yields an effective sampling rate of

$$f_s = \frac{100 \text{ MHz}}{100} = 1 \text{ MSPS}$$

per channel.

The analog front end maps the input range of $\pm 5 \text{ V}$ into the 0–3.3 V domain required by the ADC. This is achieved using an op-amp attenuator and summing configuration that biases the signal to mid-rail (1.65 V), followed by two cascaded second-order low-pass filters to suppress high-frequency noise. This scaling and level shifting allow the oscilloscope to measure typical laboratory-scale signals with both positive and negative values while maintaining the ADC operating range.

Horizontal zoom is implemented digitally in the sampler module through integer decimation of the ADC controller output rather than by reprogramming the ADC clock. At the highest sampling setting (no decimation), the system operates at the full 1 MSPS rate and can display approximately four periods of a 3 kHz sinusoidal wave across the 1280-pixel horizontal resolution. Lower effective sampling rates are obtained by decimating the ADC stream, enabling display of up to four periods of a 10 Hz waveform on the same screen. This approach keeps the ADC running continuously at a fixed 1 MSPS rate and allows decimation factors to be chosen based on the display resolution and the range of target signal frequencies, rather than repeatedly adjusting the ADC timing.

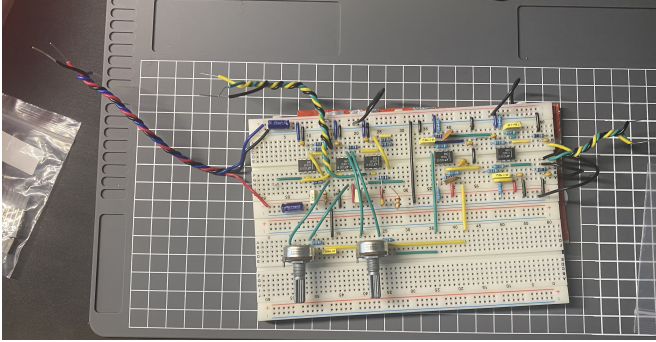


Fig. 2. Analog Front End Circuit

B. Measurements and Scaling Block

This block takes in raw channel inputs from the Data Acquisition system and calculates the signal parameters to be rendered on the display. The voltage and frequency statistics are also be used by the Bode Block for gain calculations and frequency matching verification. Among the parameters to be calculated, the most essential are be the channel V_{max} , V_{min} , average, V_{rms} and the waveform frequency.

It also support channel scaling, offset setting and binary math functions between the channels that generate the signals for the Math Channel.

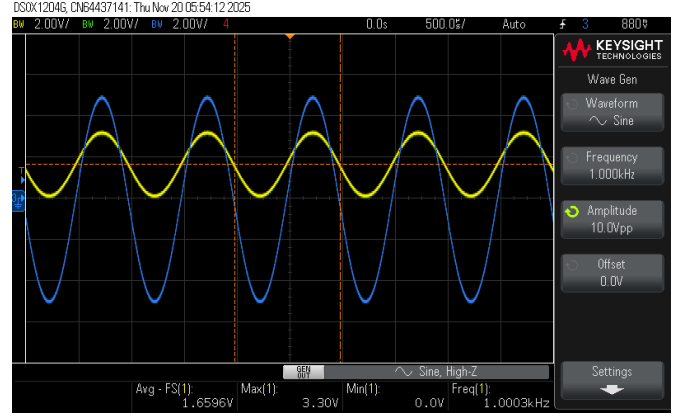


Fig. 3. Analog Front End scope shot, blue: input, yellow: output (ADC input)

C. Waveform Generation Architecture

The waveform generator is implemented using a Direct Digital Synthesis (DDS) architecture [2], whose main components are a phase accumulator, a phase to amplitude mapping stage, and a waveform selection block.

The phase accumulator is an N -bit register that is updated every clock cycle by a programmable increment phase_inc . For a system clock frequency f_{clk} , the output frequency f_{out} is given by

$$f_{\text{out}} = f_{\text{clk}} \cdot \frac{\text{phase_inc}}{2^N}, \quad (1)$$

where N is the resolution of the phase accumulator output. Larger values of phase_inc therefore produce higher output frequencies.

The most significant bits (MSBs) of the phase accumulator form a phase address that is used by the phase-to-amplitude mapping block. For sine waveforms, this block indexes a 16-bit wide and 1024 value-deep sine BRAM and returns the corresponding sample amplitude. For ramp, triangular, square, and DC waveforms, the generator bypasses the sine look-up through a multiplexer and produces samples algorithmically in the waveform core to minimize BRAM utilization and reserve the on-chip BRAMs for the storage of the various.

The DDS logic is clocked 100MHz by the system clock. A separate sample-strobe signal from the PMOD DA3 controller determines when new samples are presented to the DAC. On each strobe, the phase accumulator and waveform-generation logic advance to produce the next 16-bit output sample.

D. Data Management Subsystem

This subsystem consists of the Channel and Math buffers, XY Display Buffers, the Bode Plot storage consisting of the BODE diagram BRAM and the $\log_{10}()$ computation LUTs, wave form Lookup Tables, and the font memory used for display.

- Channel and Math Data buffers: For each of the four supported channels as well as the Math module, we allocate a pair of BRAMs organized as a ping-pong buffer. This double buffering strategy is used to facilitate

simultaneous reading and writing of the signal data. At any given time, one BRAM is in read mode, feeding data continuously to the display, while the other is in write mode, capturing new samples. Once the write BRAM is filled, the roles of the two BRAMs are reversed. This exchange ensures that the waveform is always drawn from a complete BRAM, preventing glitches during the display refresh. Each BRAM stores 1280 samples at 10-bit resolution, for 12,800 bits per BRAM. With 5 channels $\times$ 2 BRAMs/channel, the total storage for these buffers is 128000 bits.

- The XY mode uses a dense 1-bit bitmap for the full display area. Each BRAM stores a 1280 $\times$ 720 grid, one bit per pixel, so each BRAM holds 1280 $\times$ 720 = 921,600 bits. We again double-buffer (two BRAMs) so one frame can be cleared or updated while the other is being displayed. The XY store therefore requires 1,843,200 bits.
- Bode Magnitude Store: For the Bode plot, we store the interpolated gain in decibels as a 16-bit value per horizontal pixel in the Bode region. The design stores 1152 points (matching the active Bode width), so the Bode BRAM uses 18,432 bits.
- Waveform Generator Sine Lookup Table: The function generator uses a precomputed sine Look Up Table (LUT). The table has 1024 entries, each 16 bits wide, giving 16,384 bits.
- Bode Frequency ROM: The Bode controller steps through a small ROM that stores the sweep frequencies and associated pixel spacing. Each entry packs a 17-bit frequency to be sent to the waveform generator in the bode mode, a 3-bit pixel count, and a 1-bit 'is last' flag for a total of 21 bits per entry. With 24 sweep points, this ROM stores 504 bits.
- Log LUT for Gain-to-dB Conversion: The gain calculator uses a dual-channel $\log_{10}$ LUT indexed by the 12-bit V_{pp} magnitude. A full table would have 4096 addresses. Storing both channels' dB values (2 $\times$ 16 bits) per entry gives a logical table size of 131,072 bits.
- Character Sprites for Display: Text rendering for the legend uses a monochrome font ROM with 528 bytes of sprite data. This font ROM contains 66 characters, each of 528 characters each of 8 bits. The evaluates to 4,224 bits.

Memory Usage Analysis: The Urbana board provides 75 dedicated 36Kbit BRAM blocks [4]. This provides a total of 2.7Mbits of on-chip BRAM while our design requires approximately 2.32 Mbits consuming about 74.5% of the total BRAM capacity.

A summary of the total on-chip memory requirement is provided in Table I. The calculation accounts for the maximum planned usage.

E. User Control

To effectively manage the limited user interface devices available on the FPGA board, a finite-state machine (FSM) is implemented as the system's user control unit. We use four

push-buttons, sixteen slide switches, and a rotary encoder to effect the control signals. Rather than assigning each physical control to a fixed function, the design employs a menu-driven interface displayed on the seven-segment displays. Button 0 serves as the global reset, while the remaining buttons support hierarchical navigation: Button 1 (Enter) confirms selections, Button 2 (Next) advances through menu options, and Button 3 (Back) returns to the previous level. The switches and rotary encoder provide context-sensitive parameter inputs based on the active menu.

The FSM cycles through six high-level configuration states:

- **Main menu (S_MAIN):** the top-level hub entered on reset. If a Bode sweep is active, the controller remains in this state, also annotating the seven-segment display with 'BODE' and ignoring navigation requests.
- **Timebase (S_TIMEBASE):** configures the sampling rate and effects the horizontal zooming. When the user presses the enter button, the three least significant switches are latched into `sample_rate_sel`.
- **Trigger (S_TRIGGER):** configures trigger mode and trigger channel, and edits the trigger level. On enter `trigger_mode` is updated from `sw[1:0]`. Bits `sw[3:2]` select which channel's trigger level is currently controlled by the rotary encoder.
- **Channels (S_CHANNEL):** enables or disables channels, sets per-channel V/div, and adjusts DC offsets. On enter, the V/div selections for the four channels and the math output are latched from `sw[7:0]`, `ch_enable` and `math_operation` are latched from `sw[15:10]`. Bits `sw[9:8]` select which channel's offset is currently driven by the rotary encoder.
- **Signal generator (S_WAVEGEN):** configures the built-in waveform generator. On enter, the waveform type (`wave_type_sel`) is set from `sw[2:0]`, and the frequency control word (`wave_freq_sel`) is taken from `sw[15:3]`.
- **Display mode (S_MODE):** selects among normal, XY, and Bode modes. On enter, `display_mode` is latched from `sw[1:0]`, and `start_bode` is asserted for one system clock cycle when `sw[1]` is active.

Navigation between these states is implemented with a simple button protocol: the *next* button advances through TIMEBASE $\rightarrow$ TRIGGER $\rightarrow$ CHANNEL $\rightarrow$ WAVEGEN $\rightarrow$ MODE, while the *back* button returns to the main menu. The enter button does not change the state; instead, it latches the current switch configuration into the registers associated with that menu.

Continuous parameters such as trigger level and DC offset are adjusted using the rotary encoder.

Finally, the current FSM state is encoded into a 32-bit pattern that drives the seven-segment display controller. Each state is mapped to a distinct constant (e.g., BASE, HORIZONTAL, TRIGGER, CHANNEL, SIG_GEN, OPERATION), allowing the user to see at a glance which configuration page is active while reusing the same physical controls across multiple subsystems.

TABLE I
MEMORY USAGE CALCULATION FOR ENHANCED DESIGN

Item Stored	Quantity	Depth (Entries)	Bits per Entry	Total Bits	Total Memory (Kbits)
Channel & Math Ping-Pong BRAMs	10 BRAMs	1,280	10	128,000	128.000
XY Display Buffers	2 BRAMs	1280×720	1	1,843,200	1,843.200
Bode Magnitude Store	1 Table	1,152	16	18,432	18.432
Sine Wave LUT	1 Table	1,024	16	16,384	16.384
Bode Frequency ROM	1 Table	24	21	504	0.504
Font Sprites	1 Table	528	8	4,224	4.224
Total Memory Required					2,010.744 Kbits
Urbana board BRAM capacity: $75 \times 36 \text{ kbits} = 2,700 \text{ Kbits}$.					
Logical data uses $\approx 74.5\%$ of available BRAM bits.					

F. Display Block

The display block implements a digital oscilloscope interface that drives a 720p HDMI output (1280×720 at 60 Hz) using a 74.25 MHz pixel clock. It consists of four main modules: waveform display, grid, cursor control, and legend.

1) *Waveform Display Module*: For each horizontal pixel (h_count , v_count) generated by the Video Signal Generator, it sends a request to the Data Management Block to determine what color to display on the pixel and receives an answer after two cycles after internal comparison with the stored waveform data values in the dual-port BRAMs. The h_count serves as the BRAM read address, while v_count is compared against the returned voltage value to determine if a waveform pixel should be displayed at that location. To align with the 2 cycle BRAM latency, both h_count and v_count signals are pipelined through two delay stages before the final pixel color decision.

The module supports multiple display modes:

- **Normal Mode**: Waveforms for up to 4 channels
- **XY Mode**: Plots CH1 vs CH0
- **Bode Mode**: Frequency response plots (Magnitude)

When multiple display elements overlap at a single pixel, priority encoding selects the topmost element in the following order : Legend → Cursor → Waveforms → Grid → Background.

Color Assignments:

- Channel 0 (CH0): Yellow (RGB: 255, 255, 0)
- Channel 1 (CH1): Green (RGB: 0, 255, 0)
- Channel 2 (CH2): Blue (RGB: 0, 0, 255)
- Channel 3 (CH3): Magenta (RGB: 255, 0, 255)
- Math Channel: Pink (RGB: 255, 105, 176)
- Bode Plot: Purple (RGB: 128, 0, 128)
- XY Mode: Orange (RGB: 255, 160, 0)
- Grid: Grey (RGB: 63, 63, 63)
- Cursor: Orange (RGB: 255, 160, 0)
- Background: Black (RGB: 0, 0, 0)

Figure 4 demonstrates the dual-channel display capability with both channels showing DC levels. Figures 5, 6, and 7 show Channel 0 displaying sine, square, and ramp waveforms respectively, while Channel 1 maintains a DC reference level.

2) *Grid Module*: The grid module renders two grid patterns that automatically switch based on the operating mode:

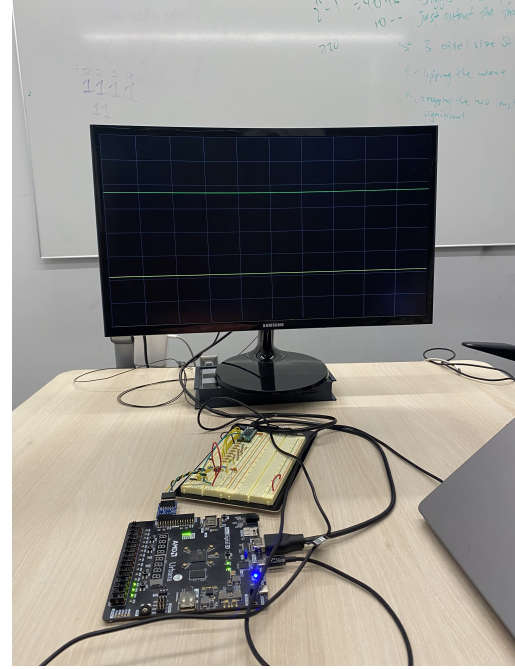


Fig. 4. Dual-channel display showing DC levels on both Channel 0 (yellow) and Channel 1 (green)

- **Normal/XY Mode Grid**: Uniform linear divisions with 10 vertical lines (128 pixels apart) and 8 horizontal lines (90 pixels apart), providing consistent spacing for time-domain and XY measurements.
- **Bode Mode Grid**: Logarithmic frequency scaling with vertical lines positioned to represent logarithmic frequency divisions covering the range from 10 Hz to 6 kHz. The grid includes two full decades (10-100 Hz, 100-1000 Hz) and a partial decade (1-6 kHz). No horizontal grid lines are displayed in Bode mode.

3) *Cursor Control Module*: The cursor module provides four independent cursors (X1, X2, Y1, Y2) for measurement purposes. User control signals using switches enable cursor selection, and rotary encoder transitions are used to move individual cursors in two directions, similar to typical oscilloscope user interfaces. The module outputs individual cursor positions (X1, X2, Y1, Y2) as well as the differences ΔX (between X1 and X2) and ΔY (between Y1 and Y2) to be displayed on the legend for time interval and voltage measurements.

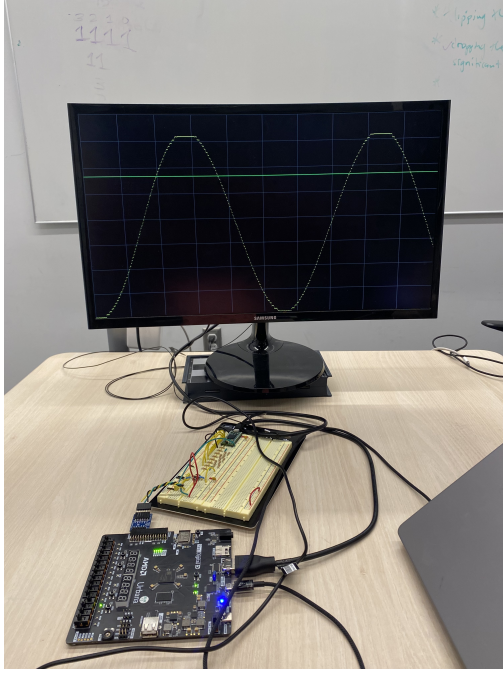


Fig. 5. Channel 0 displaying sine wave with Channel 1 DC reference

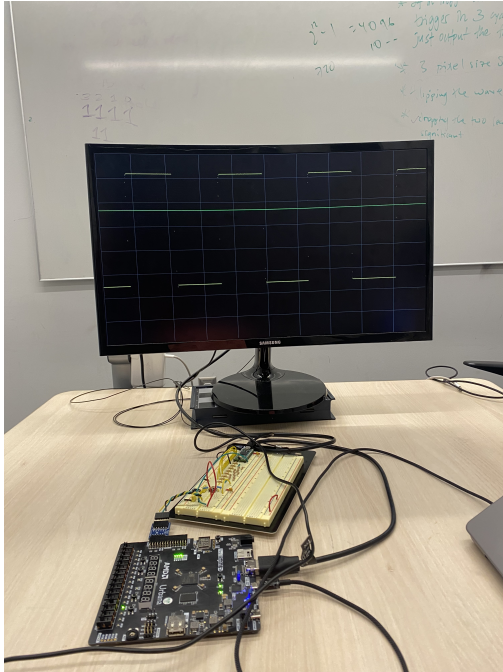


Fig. 6. Channel 0 displaying square wave with Channel 1 DC reference

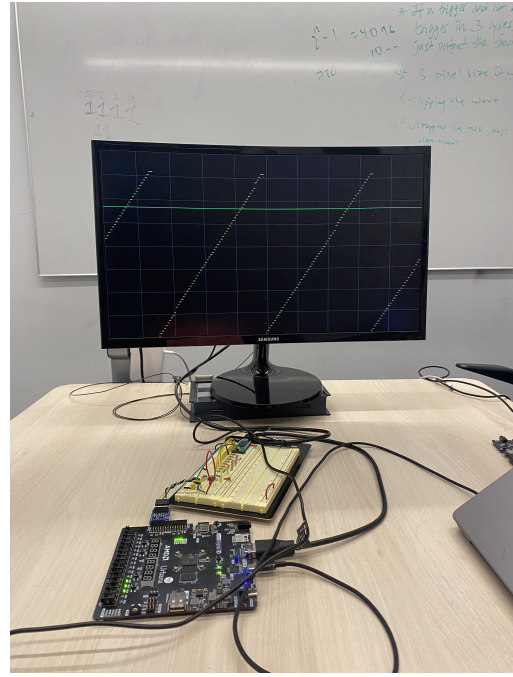


Fig. 7. Channel 0 displaying ramp waveform with Channel 1 DC reference

4) *Legend Module*: The legend module provides the user with signal parameter measurements for each active channel. This module occupies the lower screen region (vertical pixels 620–719) and displays measurement data using an 8×8 pixel character font stored in a ROM lookup table [3]. The following parameters are displayed:

- $V_{\max}$, $V_{\min}$, V_{pp} (peak-to-peak voltage)
- Frequency
- Cursor positions (X_1 , X_2 , Y_1 , Y_2) and differences (ΔX , ΔY)

III. RESULTS AND EVALUATION

A. Features and Functionality

The system successfully implements the majority of the planned digital oscilloscope features. The following summarizes the key functional components:

- **Four-Channel Waveform Acquisition**: An analog front-end conditions input signals in the $\pm 5V$ range, applying appropriate scaling and offset to map them into the 0–3.3V input range required by the ADC. All four channels are sampled simultaneously, supporting rising-edge trigger selection on any channel for stable waveform display.
- **Function Generator**: An integrated waveform generator produces sine, square, triangle, ramp, and DC signals across a 1 Hz to 80 kHz frequency range. In Bode mode, the generator automatically sweeps through logarithmically spaced frequencies (10 Hz to 6 kHz) for frequency response analysis.
- **Display Modes**:
 - *Normal Time-Domain Mode*: Displays all four channels concurrently on a 10×8 grid.

- *XY Mode*: Plots CH1 vs CH0 for phase analysis.
- *Bode Mode*: Displays gain vs frequency using a logarithmic grid.
- **Horizontal and Vertical Scaling**: The system supports both vertical and horizontal zooming to help users analyze signals at different resolutions. Vertical scaling is implemented using discrete zoom factors of $\times 1/2$, $\times 2$, and $\times 4$, allowing users to view signals with varying amplitude detail. Horizontal scaling is achieved by adjusting the effective sample rate of the system, which changes the time-per-division and thus zooms in or out along the time axis.
- **Cursors**: Four independent cursors (X1, X2, Y1, Y2) are user-controlled via a rotary encoder and switches. The system computes absolute positions and differences ΔX (time) and ΔY (voltage) between cursors for measurements.
- **Legend**: A reserved region at the bottom 100 pixels of the screen is used to display key waveform parameters in real time. These include:
 - Peak voltage values: $V_{\max}$, $V_{\min}$, V_{pp}
 - Signal frequency
 - Cursor measurements: ΔX (time), ΔY (voltage)
 - Current time/division and voltage/division scale settings
- **User Interface and Display**: The system uses switches, pushbuttons, and a rotary encoder to provide a simple, navigable menu-driven interface. Users can enable channels, switch between modes, adjust scaling, and control cursors. An eight-digit seven-segment LED display on the board is used to communicate the current menu status of the system. This provides feedback and helps users understand the system state without needing an external monitor.

B. Resource Utilization

Post-synthesis utilization results were obtained using Vivado v2025.1 targeting the XC7S50-CSGA324-1 device on the Urbana board. Table II summarizes resource usage across the major subsystems.

TABLE II
POST-SYNTHESIS RESOURCE UTILIZATION.

Resource	Used	Available	Utilization
LUTs (Total)	3,113	28,800	10.8%
LUTRAMs	26	9,600	< 1%
SRLs	22	—	—
Flip-Flops	1,803	57,600	3.1%
Block RAM (36 Kb)	62	75	82.7%
Block RAM (18 Kb)	2	—	—
DSP Slices	4	120	3.3%

a) *Subsystem breakdown.*: The Display Block (waveform, legend, and video timing) accounts for $\sim 23\%$ of total LUTs, reflecting the cost of pixel-coordinate generation and overlay logic.

The Store Block (ping-pong BRAM architecture) consumes the majority of memory resources—62 RAMB36s, or $\sim$

83% of available BRAM—primarily for channel buffers and XY/Bode memory.

The Bode Block (controller, interpolator, and log LUT) is the largest computational unit, utilizing $\sim 1.3k$ LUTs and 26 LUTRAMs for gain computation and log-domain conversion.

The Measurement and Scaling Block and Control Unit together account for ~ 400 LUTs and $< 10\%$ of total logic.

DSP48 slices ($\times 4$) are allocated to the Bode Gain Calculator and XY drawing interpolator.

b) *Summary.*: The design comfortably fits within the Artix-7 XC7S50 device, with ample logic and DSP headroom for future expansion to four-channel operation or additional math functions. BRAM usage is the dominant limiting factor, driven by framebuffer-style dual-port storage for the oscilloscope display and XY/Bode modes.

C. Latency and Throughput

System-level latency in the oscilloscope is dominated by three clock domains: the 100 MHz system clock used for acquisition and control, the 74.25 MHz pixel clock used for HDMI display timing, and the slower sample clock derived for the ADC interface (20 MHz). The pipeline from ADC sampling $\rightarrow$ BRAM storage $\rightarrow$ HDMI pixel rendering introduces an end-to-end latency of approximately 3–4 frame periods at 60 Hz (approximately 50–70 ms), primarily due to double buffering and synchronization across the data and pixel domains.

The data-acquisition path operates continuously at 1 MSPS per channel (12-bit samples). With two channels active, this yields an aggregate raw throughput of:

$$2 \times 1 \text{ MSPS} \times 12 \text{ bits} = 24 \text{ Mbps}, \quad (2)$$

which comfortably fits within the 100 MHz internal streaming bandwidth. During Bode-plot sweeps, the measurement and averaging subsystem operates at approximately 10 updates/s, limited by the function-generator frequency stepping and measurement accumulation window.

Timing analysis from post-synthesis results indicates that the design narrowly passes timing closure between the 100 MHz system clock and 74.25 MHz pixel domain. The worst-case setup slack is 1.324 ns, corresponding to a critical cross-domain path from `the_display/mvg/h_count_reg[3]` (pixel clock domain) to `new_frame_pipe_reg` (system clock domain). The three-level LUT-based combinational logic contributes approximately 1 ns of delay, while interconnect routing adds about 2 ns. This path has been further corrected through an additional synchronization stage or asynchronous FIFO at the display interface.

The rest of the design meets timing comfortably within both clock domains. The effective maximum achievable frequency ($F_{\max}$) for the 100 MHz logic is approximately 106 MHz, and 74 MHz for the HDMI pixel logic, ensuring stable 720p@60 Hz video output with deterministic sample-to-display latency. With pipelined BRAM access and block-

synchronous updates, the oscilloscope maintains real-time waveform rendering with no visible tearing or frame drop.

ACKNOWLEDGEMENTS

We would like to thank Joe for guiding us through the project, from helping us brainstorm ideas, providing the parts we needed, and always pushing us to do more. We'd also like to thank the entire 6.205 teaching staff for their support, feedback, and all the help throughout the course.

REFERENCES

- [1] <https://digilent.com/reference/pmod/pmodad1/reference-manual>
- [2] <https://www.analog.com/en/resources/analog-dialogue/articles/dds-generates-high-quality-waveforms-efficiently.html>
- [3] https://github.com/dhepper/font8x8/blob/master/font8x8_basic.h
- [4] https://fpga.mit.edu/6205/_static/F25/lectures/L08.pdf
- [5] <https://www.tequipment.net/tektronix/oscilloscopes/>