Electronics for Embedded Systems

Experimental Project

Fabio Scatozza (315216)

July 21, 2025

Contents

References

[1] Analog Devices. *AD7928 8-Channel, 1 MSPS, 12-Bit ADC with Sequencer in 20-Lead TSSOP*. Datasheet. Version E. URL: https://www.analog.com/media/en/technical-documentation/data-sheets/AD7908_7918_7928.pdf.

[2] Anhui Vico Technologes. *Metal Film Fixed Resistors*. Datasheet. Dec. 2019. URL: https://www.lcsc.com/datasheet/lcsc_datasheet_2409272302_VO-MF2WS-0R68-1--6T63_C2903178.pdf.

[3] IBM Corporation. *Personal System/2 Hardware Interface Technical Reference*. Oct. 1990. Chap. Keyboard and Auxiliary Device Controller. URL: http://www.mcamafia.de/pdf/ibm_hitrc07.pdf.

[4] IBM Corporation. *Personal System/2 Hardware Interface Technical Reference*. Oct. 1990. Chap. Keyboard 101- and 102-Key. URL: http://www.mcamafia.de/pdf/ibm_hitrc11.pdf.

[5] Intel Corporation. *Embedded Memory (RAM: 1-PORT, RAM: 2-PORT, ROM: 1-PORT, and ROM: 2-PORT) User Guide*. URL: https://www.intel.com/content/www/us/en/docs/programmable/683240/17-0/about-embedded-memory-ip-cores.html.

[6] Intel/Altera Corporation. *Cyclone V SE SoC FPGA*. URL: https://www.intel.com/content/www/us/en/products/details/fpga/cyclone/v/se.html.

[7] Linear Technology Corporation. *LTC2308 Low Noise, 500ksps, 8-Channel, 12-Bit ADC*. Datasheet. Version C. URL: https://www.analog.com/media/en/technical-documentation/data-sheets/2308fc.pdf.

[8] Mean Well Enterprises. *LRS-50-12 50W Single Output Switching Power Supply*. Datasheet. Feb. 2024. URL: https://www.meanwell-web.com/content/files/pdfs/productPdfs/MW/LRS-50/LRS-50-spec.pdf.

[9] Stepperonline. *17HE15-1504S E-Series Nema 17 Bipolar Stepper Motor*. Datasheet. May 2023. URL: https://www.omc-stepperonline.com/index.php?route=product/product/get_file&file=2838/17HE15-1504S_Full_Datasheet.pdf.

[10] STMicroelectronics. *L298 Dual Full-Bridge Driver*. Datasheet. Oct. 2023. URL: https://www.st.com/resource/en/datasheet/l298.pdf.

[11] STMicroelectronics. *L6505 Current Controller For Stepping Motors*. Datasheet. July 2003. URL: https://www.st.com/resource/en/datasheet/l6506.pdf.

[12] STMicroelectronics. *Nucleo-64 Development Board with STM32F401RE MCU*. Datasheet. Jan. 2015. URL: https://www.st.com/resource/en/datasheet/stm32f401re.pdf.

[13] Terasic. *DE1-SoC Board*. URL: https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=165&No=836.

[14] Vishay General Semiconductor. *SB260 Schottky Barrier Plastic Rectifier*. Datasheet. Apr. 2020. URL: https://www.vishay.com/docs/88717/sb220.pdf.

[15] XSD. *6063-T5 Aluminum Heatsink*. Datasheet. Mar. 2017. URL: https://www.lcsc.com/datasheet/lcsc_datasheet_1811100911_XSD-30-30-25-2P_C32487.pdf.

1 Introduction

The *Electronics for Embedded Systems* course explores the analog and digital aspects of embedded system design, focusing on the key components of their typical architecture. While sensor and actuator specifics are addressed in other courses, this course emphasizes the interface between the analog and digital domains, particularly A/D and D/A conversion, as well as the power electronics of actuator drivers and power supply devices. In the digital domain, the course spans software implementations running on microprocessors to hardware synthesized on programmable devices, with an emphasis on the underlying technology enabling computation, communication, and storage. Specifically, this involves analyzing the theory of programmable logic devices, microprocessor embedded peripherals, wired communication protocols, and memories.

In this context, the document presents a small-scale project aimed at addressing the key topics of the course, demonstrating competency aligned with the expected learning outcomes.

2 Theory of Operation

The system implements an interface to program the movement pattern for a bipolar stepper motor and execute it on demand. The pattern is committed to non-volatile storage and consists of a sequence of data points, each representing a circular motion segment defined by angular velocity and displacement. During execution, the data points are retrieved in a circular fashion from memory, and are converted into motor-driving stimuli. Referring to the system block diagram in fig. 1, the interface is operated as follows.

- The movement pattern is cleared by pressing and holding the user push button for at least 1 s. Successful clearing is acknowledged by displaying *CLEAr* on the 7-segment displays.
- Data points are entered using the linear potentiometer and the PS/2 keyboard. The potentiometer sets the angular velocity in the range from 2.5 rpm to 400 rpm, while the angular displacement is specified in degrees with the keyboard, according to the following regex pattern.

  ```
  [+-]?[0-9]{1,3}(\.[0-9])?
  ```

  The sign defaults to `+` if omitted, which corresponds to a counterclockwise rotation. The integer part is specified with up to three digits, whereas the fractional part is optionally specified with one digit. Due to the inherently discrete nature of the stepper motor rotation and accounting for the two-phases-on driving method, the target angular displacement is rounded to the nearest value that minimizes error.

  When the data point entry begins, triggered by pressing some keys, the 7-segment displays change to *InPUt*. The operation is terminated by pressing the return ↵ or Enter keys, upon which data is validated, and the potentiometer output is converted. The outcome is notified as follows: *Err-2 bAd PAttErn*, if the parsing of the keyboard input failed; *Err-3 FuLL*, if the maximum number of data points have already been entered. Otherwise, if no error has occurred, the data point is displayed in the format below, where the data point number starts at zero, the angular velocity is expressed in revolutions per minute, and the angular displacement in degrees.

  ```
  [<data point number>] <angular velocity> <angular displacement>
  ```

- The execution of a valid movement pattern, that is, one containing at least one data point, begins by pressing the user push button. An empty pattern results in the display of *Err-1 no dAtA*, while a valid pattern displays *PLAy*. The execution continues until the push button is pressed again, after which the displays revert to printing *IdLE*.

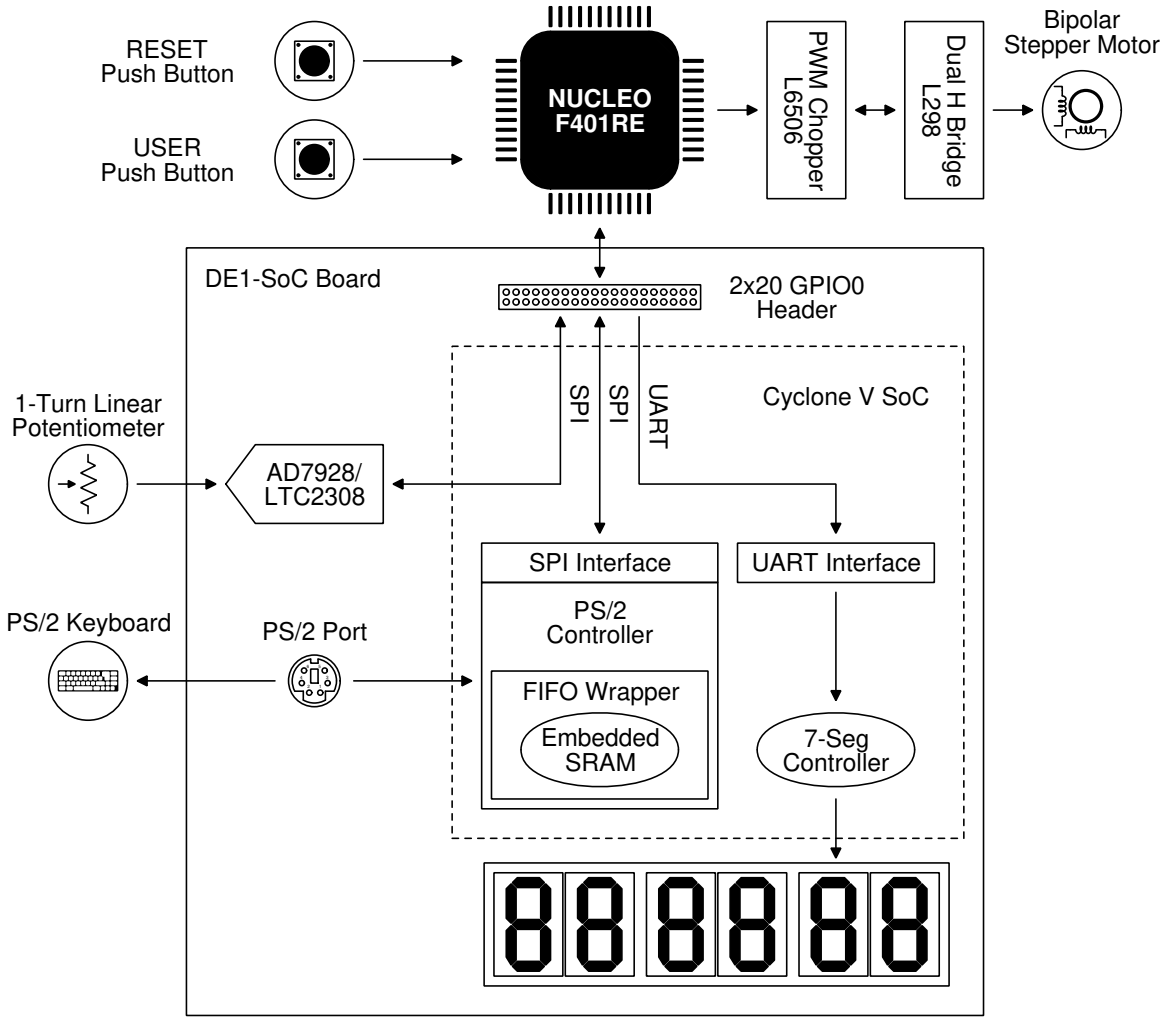
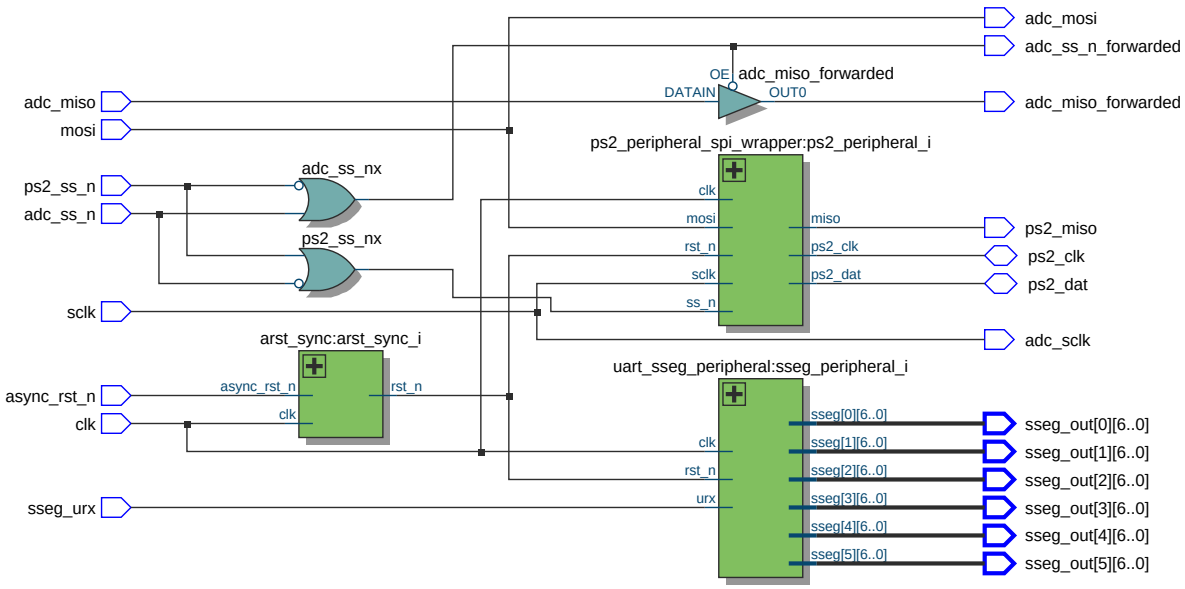
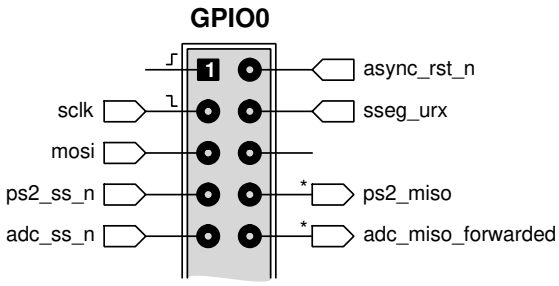


Figure 1: System block diagram



(a) Quartus Prime RTL view



(b) Signals mapped to the 2x20 header for interfacing with the MCU

Figure 2: DE1-SoC board subsystems

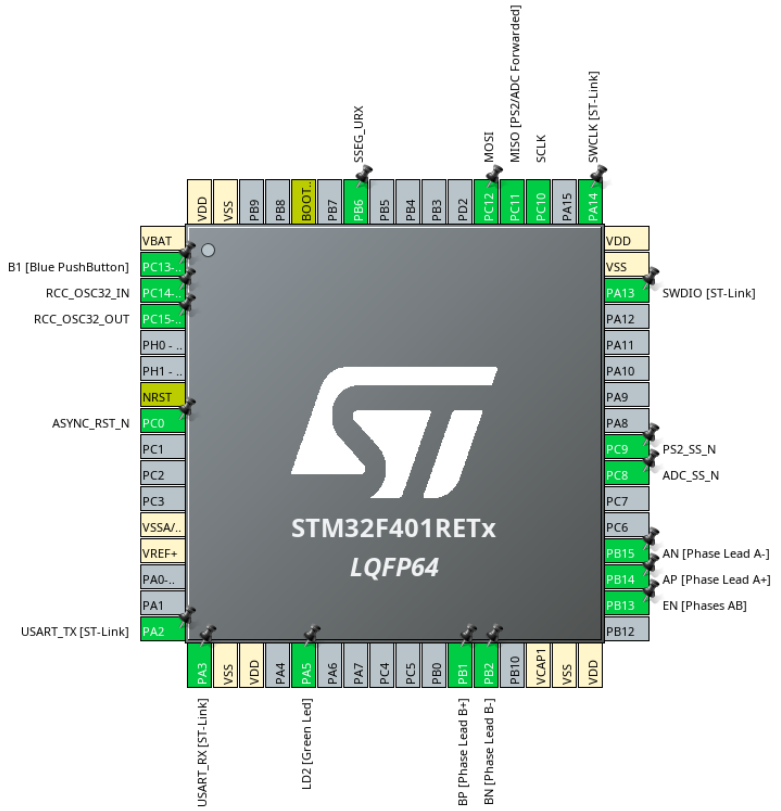


Figure 3: Microcontroller pinout

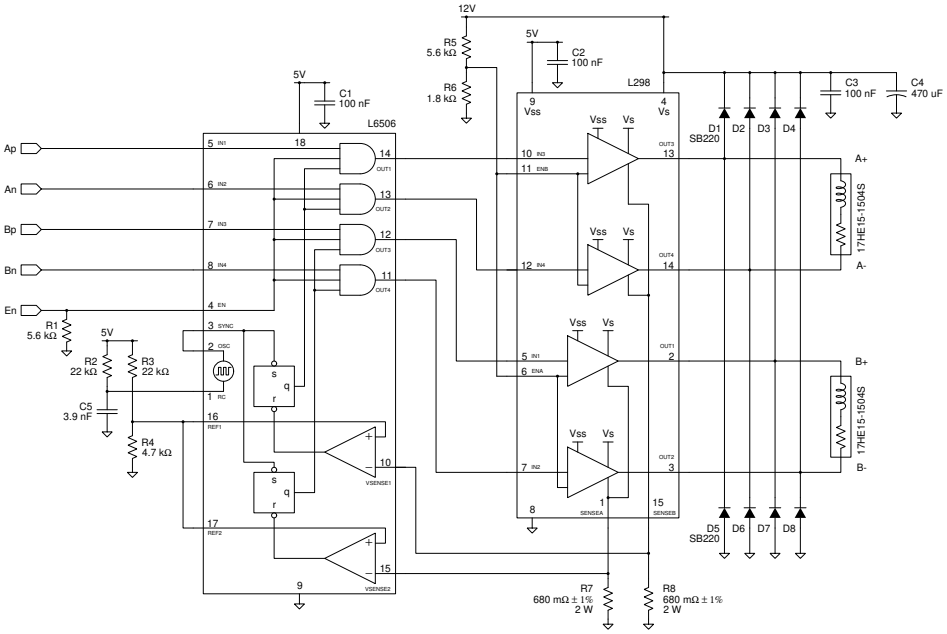


Figure 4: Drive circuit for the bipolar stepper motor. The L6506 senses and controls the current in the motor windings, which are energized with the dual full-bridge driver L298.

3 System Design

The system is devised to target the hardware available on the design platform employed for the laboratory activities of the course, the DE1-SoC development kit by Terasic [13]. The platform is built around Altera's Cyclone V SE SoC device [6], which integrates an ARM-based Hard Processor System (HPS) tied to the FPGA fabric with an on-chip interconnect. Potentially, this kit alone allows to experiment with digital processing in the form of software running on a hard or soft microprocessor to program hard and soft peripherals, as well as by synthesizing dedicated hardware. Despite this flexibility, the processor system is implemented with an external microcontroller development board, the NUCLEO-F401RE by STMicroelectronics [12], thus creating the opportunity to investigate communication protocols for wired interconnections. As illustrated in the block diagram of fig. 1, the system comprises:

- Input devices.
  - The user and reset push buttons are part of the microcontroller development board.
  - The linear potentiometer is wired to the ADC available on the DE1-SoC board, through the dedicated 2x5 header. This is discussed in section 3.1.
  - The PS/2 keyboard is attached to the corresponding port on the DE1-SoC board, through the fitted 6-pin mini-DIN connector.
- Peripherals and glue logic implemented on the FPGA target, as more accurately shown in fig. 2a. The interface to the microcontroller consists of the signals mapped to the 2x20 expansion header GPIO0, as depicted in fig. 2b.
  - The SPI-compatible port of the converter is routed through the FPGA fabric, up to the GPIO0 header.
  - An asynchronous reset signal driven by the microcontroller is brought into the FPGA 50 MHz clock domain, to be used as a synchronously deasserted system-wide reset.
  - The PS/2 peripheral implements the PS/2 protocol to communicate with the keyboard device, while it exposes an SPI port to the microcontroller. The design of the module is presented in section 3.2.
  - The display peripheral drives the 7-segment displays available on the DE1-SoC board with the character stream received from the microcontroller via UART. This is discussed in section 3.3.

  The slave select inputs are internally pulled up, to ensure an inactive level while the microcontroller resets.

  The blocks are modeled in SystemVerilog and are functionally simulated with testbenches that make use of the verification subset of the language. The approach to verification is directed testing, with non-constrained randomized stimuli. As a result, the testbenches can be run with the freeware edition of Questa Intel Starter FPGA Edition. The synthesis flow is carried out with the Quartus Prime Lite toolchain, using bash and Tcl automation scripts.
- Microcontroller and bipolar stepper motor drive circuit, as illustrated in greater details in figs. 3 and 4. The microcontroller development board, the L6506 controller [11], and the logic section of the L298 driver [10], are powered at 5 V from the DE1-SoC board; the microcontroller power supply is further regulated down to 3.3 V.
  - As the FPGA I/Os are configured in the 3.3 V LVTTL standard, the microcontroller directly interfaces to the signals on the 2x20 header GPIO0. Notably, the PS/2 and ADC MISO signals are wired as a shared line into the microcontroller, and the input is internally pulled down to ensure a well-defined logic level while neither slave is selected.

    Several peripherals are configured to achieve the desired functionality, most importantly:

  - The GPIO, to detect push button presses, as well as for generating reset and enable signals for other subsystems.
  - The SPI3 and USART1, to communicate with the peripherals synthesized in the FPGA.
  - The TIM1, to generate two pairs of complementary PWM waveforms, with one pair in quadrature to the other, to drive the bipolar stepper motor.

  Furthermore, one sector of the embedded NOR flash memory is used as a non-volatile circular buffer to store the motor movement pattern, and play it on demand.
- The L6506 inputs accept 5 V TTL levels, with $V_{\mathrm{IL}} = 0.8\,\mathrm{V}$ and $V_{\mathrm{IH}} = 2\,\mathrm{V}$. Consequently, the microcontroller can reliably drive them, without requiring any additional translation logic. During reset, the enable signal is pulled down with an external resistor and forces a motor brake.
- The L298 is permanently enabled. The datasheet prescribes that the enable signals of the H-bridges are to be driven low before turning on or off the power stage, which is achieved by means of the resistive divider (R_5, R_6).

The design of the circuit, for the bipolar stepper motor 17HE15-1504S [9], is discussed in section 3.4.

3.1 Angular Velocity Acquisition

Depending on the board revision, the ADC is either the AD7928 [1] or the LTC2308 [7], which feature 12 bit of resolution with a successive approximation topology, and an SPI-compatible serial interface. The analog input range changes slightly between the two devices: for the AD7928, the full range can be digitally selected equal to the external 2.5 V nominal reference or double its value. For the LTC2308 instead, it is fixed to 4.096 V. In either case, the 2x5 dedicated header only provides access to the eight input channels, to the ground, and to the 5 V supply rail, thus preventing measurements in a ratiometric configuration.

Accordingly, for the AD7928, a 10 kΩ linear potentiometer is wired between the 5 V supply rail and ground, with the wiper node connected to a single-ended input channel. The 500 µA nominal current through the voltage divider makes the 1 µA leakage current negligible. In the worst case scenario for the sample and hold circuitry, the wiper of the potentiometer sits at halfway position, such that the output resistance of the thevenin equivalent R_{s} is at its maximum value of 2.5 kΩ. In this condition, with a typical hold capacitance C_{h} of 20 pF and an equivalent input resistance R_{in} of 400 Ω, a fully discharged hold capacitor charges up to the 2.5 V input V_{s} following

$$v_{\mathrm{h}}(t) = V_{\mathrm{s}} \left(1 - e^{-t/\tau}\right), \qquad \text{with} \qquad \tau = (R_{\mathrm{s}} + R_{\mathrm{in}})\, C_{\mathrm{h}}$$

and the time required for this voltage to settle within half LSB of the final value is

$$t_{\mathrm{set}} = \tau \, \ln \frac{V_{\mathrm{s}}}{0.5\,\mathrm{LSB}} \tag{1}$$

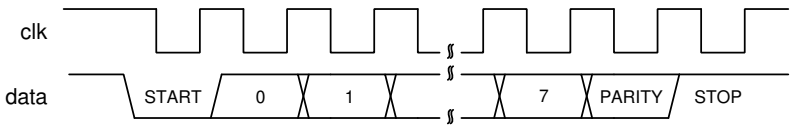
Hence, $t_{\mathrm{set}} = 482\,\mathrm{ns}$ bounds the minimum acquisition time for achieving full resolution conversions.

Considering the narrower analog range of the LTC2308, the 10 kΩ potentiometer is wired at the bottom of a voltage divider, with the resistor R_1 on the upper branch sized to limit the voltage drop on the potentiometer. Denoting with V_{s} the supply voltage and with V_{fr} the ADC full range, it must hold

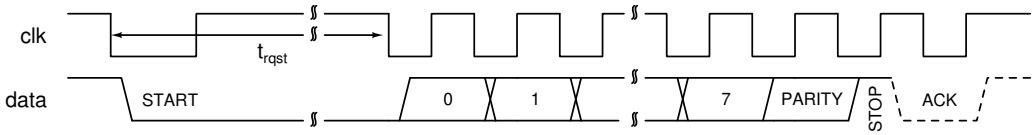
$$V_{\mathrm{s}} \frac{R_{\mathrm{pot}}}{R_{\mathrm{pot}} + R_1} \leq V_{\mathrm{fr}}$$

Therefore, in the typical case for the supply and full range voltages, but considering a 10 % uncertainty on the potentiometer resistance and a 1 % uncertainty for R_1, it is required that

$$R_{1,\mathrm{min}} \geq R_{\mathrm{pot,max}} \left(\frac{V_{\mathrm{s}}}{V_{\mathrm{fr}}} - 1\right)$$



(a) System receiving data. The device generates data while the clock is high.



(b) System sending data. The system generates data while the clock is low; following the parity bit, the device detects the stop bit and pulls the data line low while the clock is high (dashed portion of the waveform).

Figure 5: PS/2 timing diagrams. The clock period ranges from 60 µs to 100 µs; the maximum request-to-send duration is 15 ms.

which yields $R_1 \geq 2.43\,\text{k}\Omega$. Accordingly, R_1 can be chosen in the E12 series with a nominal value of $2.7\,\text{k}\Omega$. Finally, the observation on the minimum acquisition time for the AD7928 applies here too. Given the different configuration of the voltage divider, the output resistance is maximized when the wiper reaches a position where the total divider resistance is split in half, hence R_s is $3.18\,\text{k}\Omega$. Considering the equivalent model of the analog input, with a typical hold capacitance C_h of 55 pF and an equivalent input resistance R_in of $100\,\Omega$, eq. (1) yields $t_\text{set} = 1.53\,\mu\text{s}$.

The four signals of the ADC serial interface are routed through the FPGA fabric up to the 2x20 GPIO0 expansion header, where they connect to the microcontroller pins assigned to the SPI peripheral. The peripheral is configured as a full-duplex master, with software-controlled slave select signals.

3.2 Angular Displacement Acquisition

The PS/2 protocol [3, 4] regulates the point-to-point bidirectional communication between the system and the keyboard device. It is a synchronous serial protocol in which data is time-multiplexed over a single data line, while a dedicated line carries the clock signal. Accordingly, the communication is half-duplex and both clock and data lines are shared with an open-drain architecture to avoid damage in case of conflicts.

The clock is always generated by the PS/2 device, but the system can pull the clock line low to inhibit the bus, potentially causing collisions that are used to abort ongoing transmissions in either direction. When the bus is not inhibited, two configurations are possible prior to a transmission: if the data line is also pulled high, the bus is in its idle state, which is the only one in which the device can initiate a transmission to the system. Alternatively, if the data line is held low by the system, the configuration signals the device that the system is ready to transmit data and awaiting clock generation. The transmission is character-based and the frame comprises:

- Start bit, always 0
- 8 bit character, transmitted LSB first
- Odd parity bit, either 0 or 1 such that the overall count of bits with value 1 in the character concatenated with the parity bit is odd
- Stop bit, always 1

The timing diagram for receiving and sending data are shown in fig. 5.

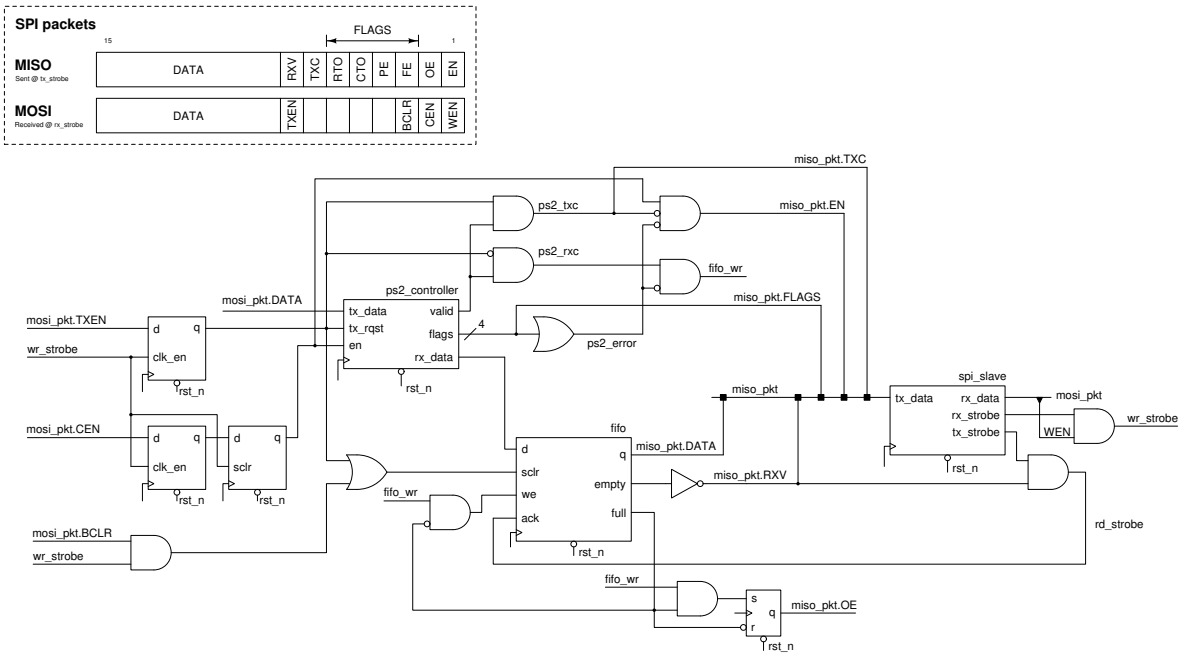


Figure 6: SPI-compatible PS/2 controller peripheral design. The module implements the PS/2 protocol for exchanging data with a PS/2 device, exposing its control, status, and data registers through SPI. Data received from the device is buffered in an SRAM FIFO. The PS/2 and SPI buses are not shown.

3.2.1 PS/2 Peripheral

The PS/2 protocol is implemented by the custom peripheral shown in fig. 6, which enables data exchange with an external PS/2 device, under the control of an SPI-capable system. It includes three key modules: a PS/2 controller, a show-ahead single-clock FIFO, and an SPI slave controller. Additional glue logic implements the desired behavior for the control and status register bits that are exposed through SPI.

PS/2 Controller It implements the PS/2 protocol specifications and provides an higher-level simplified interface to the PS/2 device, as presented in listing 3.1. It supports bidirectional communication and provides several reliability features. The data and clock lines are synchronized to the system clock domain with a flip-flop chain: the depth defaults to 2, which achieves a reasonable trade-off of latency and Mean Time Between Failures (MTBF) due to metastable events. The controller detects:

- Frame errors. On reception, if the start or stop bit is missing at the expected time. On transmission, if the acknowledgment bit is similarly absent.
- Parity error, if the parity check fails during reception.
- Clock timeouts, if the falling edges of the clock are separated by more than 150 µs during either reception or transmission.
- Request timeouts. On transmission, if the device does not generate the clock within a 15 ms window after the system inhibits the PS/2 bus (see fig. 5).

The controller has been designed according to the ASM methodology, starting from a pseudocode description of the receive and transmit operations; the resulting ASM chart and data path are provided in appendix A. In summary, the module operates as follows.

```
module ps2_controller #(
  parameter int unsigned SYNC_STAGES = 2,
  parameter real FCLK_HZ = 50e6,
  parameter int unsigned TIM_PSC = 249 // tim_fclk = fclk/(PSC+1) = 200 kHz, for 50 MHz clock
) (
  input var logic clk,
  input var logic rst_n,
  input var logic en,

  input var logic tx_rqst,
  input var logic [7:0] tx_data,

  output logic valid,
  output logic [7:0] rx_data,
  output ps2_pkg::flags_t flags,

  inout tri ps2_clk,
  inout tri ps2_dat
);
```

Listing 3.1: SystemVerilog interface of the PS/2 controller. The number of flip-flop synchronizing stages for the PS/2 lines is parameterized and defaults to 2. The timing thresholds are computed at compile-time, based on the specified clock frequency and prescaler value.

- On reset or while not enabled, the controller inhibits the bus by driving the clock line low.
- While enabled, with no pending transmission request, the controller monitors the bus for transmissions from the device. In case of successful reception, the valid signal pulses for one clock cycle and the received character appears on the data output. Conversely, in case of failure, the valid signal is held high while the flags signal the cause. While in this failure state, the PS/2 bus is inhibited; resuming normal operation requires a falling transition on the enable signal.
- While enabled, with no ongoing transmission to the device, the controller samples the data to be transmitted in the same clock cycle where the transmission request gets asserted. In case of successful transmission, the valid signal is held high until the request signal becomes inactive, and the bus is inhibited. Conversely, the failure behavior is the same as in reception.

Show-Ahead Single-Clock FIFO Data received from the PS/2 device is buffered in a synchronous FIFO structure with a show-ahead interface. Provided that the empty signal is inactive, this interface automatically outputs the first valid word, and asserting the read-acknowledge signal causes the FIFO to output the next valid word.

The module is designed to infer embedded dual-port SRAM blocks if available, without depending on a specific read-during-write behavior. On reset, the read and write pointers are cleared, and the FIFO is empty. With the first write operation, the data is stored at the write pointer location, which is the same one being read: due to the read-during-write event, the SRAM may output the old data, thus the latency for the data to appear at the output is 2 clock cycles. Accordingly, the empty flag is cleared with the same latency. The full design is provided in appendix B.

SPI Slave Controller It implements the SPI protocol in the slave role, with programmable frame length, clock polarity (CPOL) and clock phase (CPHA).

To decrease clock oversampling requirements, the receive and transmit logic operate within the domain of the external SPI clock, generated by the master device; the slave select signal crosses into the system clock domain with a dual flip-flop synchronizer. While the slave is not selected,

the data to be transmitted is sampled in the system clock domain up to one clock cycle before the transmit strobe signal pulses high, which marks the start of communication. At the end of the communication, when the slave is deselected, the received data is loaded onto the data output and becomes valid in the system clock cycle when the receive strobe signal pulses high. Appendix C contains the design of the module and the manual timing analyses.

The peripheral exposes its control, status, and data registers as packets exchanged through SPI, according to the format summarized in fig. 6. A detailed description of the packets format is provided below.

MOSI Packet

Bit 0 **WEN**: Write Enable
- 0: The packet is discarded
- 1: The packet is decoded

Bit 1 **CEN**: Controller Enable
- 0: Disables the controller and inhibits the PS/2 bus (clock line low)
- 1: Clears the flags (FE, PE, CTO, and RTO), and enables the controller in the operation mode set by TXEN.
  If the controller is already running (EN bit is set), the current operation is aborted. Aborting a receive operation while the PS/2 device is transmitting a character generates a communication error (FE, PE, or CTO set).

Bit 2 **BCLR**: Buffer Clear
- 0: The receive FIFO buffer is not flushed
- 1: The receive FIFO buffer is flushed. If both CEN and BCLR are set, the flush operation is performed prior to enabling the controller.

Bit 7 **TXEN**: Transmitter Mode Enable
This bit determines the operation mode.
- 0: The controller receives data from the PS/2 device. Incoming packets are stored in the FIFO buffer, whose state is represented by the RXV and OE status bits. When the FIFO is empty, the RXV bit is cleared and the DATA field is undefined. Otherwise, the RXV bit is set, and the DATA field is valid. If the FIFO buffer is full and new packets are received, the OE status bit is set. Each SPI transfer removes one element from the FIFO, which may also clear the OE bit. Alternatively, the OE bit is cleared by flushing the FIFO buffer when the BCLR control bit is set.
- 1: The FIFO buffer is cleared, and the controller sends data to the PS/2 device. When the operation completes, the TXC bit is set, the flags are updated (FE, CTO, and RTO), the EN bit is cleared and the PS/2 bus is inhibited (clock line low).

Bits 15:8 **DATA**: Transmit Data
If TXEN is set, the data to be sent to the PS/2 device.

MISO Packet

Bit 0 **EN**: Controller Enabled
This bit is set by the master when enabling the controller and is cleared by the hardware in one of the following cases:

- The communication has failed and a flag is set (FE, PE, CTO, or RTO).
- The TXEN control bit enabled the controller in transmitter mode, and the transmission has terminated (TXC is set).

With the EN bit cleared, the PS/2 bus is inhibited (clock line low).

Bit 1 **OE**: Overrun Error

This bit is set by the hardware when the FIFO buffer is full and a new packet has been received and lost. It is automatically cleared when removing elements from the FIFO with SPI transfers, when flushing the buffer by setting the BCLR bit, or when the controller is enabled in transmitter mode.

Bit 2 **FE**: Frame Error

This bit is set by the hardware if the communication has failed in one of the following ways. While receiving data from the PS/2 device, either the START or STOP bits are not recognized at the expected time. While transmitting data, the ACK bit is not recognized at the expected time.

Once the error is raised, the controller halts and inhibits the PS/2 bus (clock line low). Resuming normal operation requires enabling the controller again.

Bit 3 **PE**: Parity Error

This bit is set by the hardware if the parity check on the data received from the PS/2 device has failed.

Once the error is raised, the controller halts and inhibits the PS/2 bus (clock line low). Resuming normal operation requires enabling the controller again.

Bit 4 **CTO**: Clock Timeout

This bit is by the hardware if the falling edges of the clock generated by the PS/2 device are separated by more than 150 µs, during either reception or transmission.

Once the error is raised, the controller halts and inhibits the PS/2 bus (clock line low). Resuming normal operation requires enabling the controller again.

Bit 5 **RTO**: Request Timeout

This bit is set by the hardware if, on transmission, the PS/2 device has failed to generate the clock within a 15 ms window after the PS/2 bus was inhibited.

Once the error is raised, the controller halts and inhibits the PS/2 bus (clock line low). Resuming normal operation requires enabling the controller again.

Bit 6 **TXC**: Transmission Complete This bit is set by the hardware when the controller was enabled in transmitter mode and the transmission has completed (with or without errors).

Bit 7 **RXV**: Receive Data Valid This bit is set by the hardware when the DATA field is valid and contains one character received from the PS/2 device. The read operation from the FIFO is automatically acknowledged.

Bits 15:8 **DATA**:

If RXV is set, one character received from the PS/2 device.

3.3 7-Segment Displays Peripheral

The display peripheral is devised to serially control a set of 7-segment displays. It processes a stream of UART characters, where each character can either instruct the peripheral to clear all displays or update the input for a single display; one character after the other, the displays can be all updated in a rotating

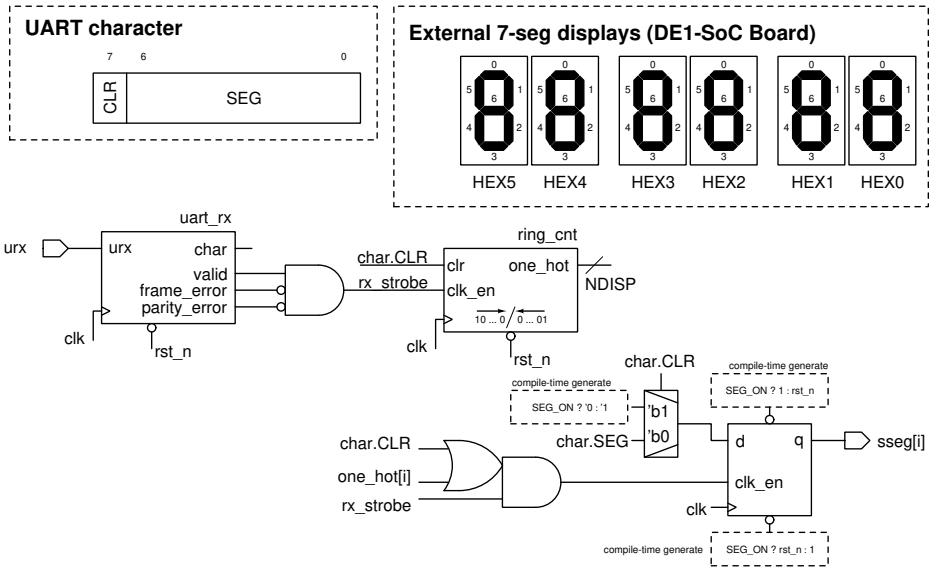


Figure 7: Display peripheral design. The module circularly displays an UART character stream on a set of 7-segment displays.

manner. The design, shown in fig. 7, makes use of a custom UART receiver module. When a transmission completes without frame or parity errors, the character appears on the output while the valid signal pulses high: the most significant bit of the character is decoded as the clear command. If the bit is set, the registers driving the display segments are forced into the off-state specified at compile-time, and the one-hot counter is cleared. Otherwise, the one-hot counter state is used to mask all clock-enable signals of the registers driving the display segments, except for one. This register is loaded with the received character data, while the one-hot counter shifts to point to the next display in the set.

3.3.1 UAR(T) Core

The core implements a flexible asynchronous serial receiver for the NRZ-L encoding. It supports several compile-time programmable parameters:

- Frame format:
  - Character length
  - Presence and type of parity (even or odd)
  - Number of STOP bits
- Baud rate, by specifying system clock, prescaler, and target clock frequency
- Serial line synchronizer depth, which determines the trade-off between data latency and MTBF due to metastable events

In addition, it features clock synchronization on all transitions, besides the UART-mandated synchronization on the START bit. Upon reception, the valid signal pulses high for one clock cycle, while the received character and error flags are latched on the outputs.

The module has been designed according to the ASM methodology, starting from the pseudocode in listing 3.2; the resulting ASM chart and data path are provided in appendix D.

```
frame[1 + NCHAR + PARITY_NONE?0:1 + NSTOP] = {1, ..., 1};
pbit = PARITY_ODD ? 1 : 0;
tim_reload_val = HALF_BIT_TIME;
tim_auto_reload_val = BIT_TIME;

while(!urx_negedge);
tim_reload();

do {
  // wait sampling instant
  while(!tim_tc) {
    // if there are transitions, resync clocks
    if (urx_edge)
      tim_reload();
  }
  // sample
  frame = {urx_sync, frame[MSB:1]};
  // update parity (start, parity xor'd too)
  if (!PARITY_NONE & (&frame[NSTOP-1:0]))
    pbit ^= urx_sync;

} while (frame[0]);

frame_error = ! &frame[MSB -: NSTOP];
parity_error = PARITY_NONE ? 0 : pbit;
rx_data = frame[1 +: NCHAR];
valid = 1;
```

Listing 3.2: Pseudocode of the asynchronous receive operation for the ASM design metodology

3.4 Bipolar Stepper Motor Drive Circuit

For educational purposes, the drive circuit does not employ fully integrated bipolar motor drivers. Instead, the L298 integrates two full-bridges to energize and reverse the phase currents. Additionally, the L6506 addresses the limitations of the simpler L/nR drive topology, in which the time constant of the inductive load is reduced by adding external resistances in series with the coils, while the desired asymptotic current is restored by increasing the drive voltage. To avoid the large power dissipation of the external resistances, the approach involves increasing the slew rate of the phase currents with an higher drive voltage, and limiting the phase currents by switching the driver. As shown in fig. 4, the phase current is sensed on the return path from the H-bridge. During the low time of the oscillator period, the drive signals pass through. However, if the phase current exceeds the limit set by the reference voltage, the drive signals are gated for the remaining part of the period. The energy stored in the inductor is dissipated as the current recirculates in the lower half of the bridge: through one snubber diode, the coil resistance, the transistor in the opposite branch, and the sense resistance. Notice that, being possible to independently set the peak phase currents, this chopping circuit supports a microstepping drive technique.

The power output stage of the L298 operates from a fixed 12 V switching power supply [8]. By means of the (R_5, R_6) divider, the enable signal pulls low as the supply voltage is about to turn on or off [10, p. 9]. Considering the input characteristics and accounting for a sensible noise margin, it must hold:

$$V_{\mathrm{IH}} + N_{\mathrm{MH}} \leq V_{\mathrm{s}} \frac{R_6}{R_5 + R_6} \leq V_{\mathrm{ss}}$$

$$\frac{V_{\mathrm{s}}}{R_5 + R_6} \gg I_{\mathrm{IH}}$$

To ensure that the enable signal pulls low in the widest range of the voltage supply dynamic, and assuming

a 5 % tolerance on the resistances, the nominal values in the E12 series can be chosen as:

$$R_5 = 5.6\,\mathrm{k\Omega} \qquad R_6 = 1.8\,\mathrm{k\Omega}$$

The bipolar stepper motor 17HE15-1504S is rated for a phase current of 1.5 A, and is characterized by a coil resistance of $(2.30 \pm 0.23)\,\Omega$ and a coil inductance of (4.0 ± 0.8) mH. Working with a 10 % derating factor, the current driven by the L298 is limited to 1.35 A per channel: in the worst case of a total saturation voltage $V_{\mathrm{CEsat}} = 4.9\,\mathrm{V}$, the total power dissipated by the driver amounts to 13.23 W. Now, for the multiwatt15 package in free air, the temperature at the junction would rise above the ambient temperature by:

$$\Delta T = R_{\theta\mathrm{ja}}\, P_{\mathrm{tot}} \approx 463\,^{\circ}\mathrm{C}$$

hence, the need to lower the thermal resistance with an heat sink. Common heat sinks for this form factor are made of 6063-T5 aluminum alloy, which has a typical thermal conductivity κ of 209 W/mK (for instance, [15]). From the mechanical data of the package, the minimum contact area S to the heat sink is $338.1\,\mathrm{mm}^3$; therefore, the thermal resistance across the contact area, for an average thickness Δl of 5.52 mm is:

$$R_{\theta\mathrm{sa}} = \frac{\Delta l}{\kappa S} \approx 78.2 \times 10^{-3}\,^{\circ}\mathrm{C/W}$$

Finally, by using a thermal compound to improve the thermal coupling between the case and the heat sink, it is reasonable to achieve:

$$R_{\theta\mathrm{cs}} \approx 0.5\,^{\circ}\mathrm{C/W}$$

In conclusion, the driver is expected to work at a maximum ambient temperature of:

$$T_{\mathrm{a,max}} = T_{\mathrm{j,max}} - (R_{\theta\mathrm{jc}} + R_{\theta\mathrm{cs}} + R_{\theta\mathrm{sa}})\, P_{\mathrm{tot}} \approx 83\,^{\circ}\mathrm{C}$$

which covers the broadly accepted range for consumer electronics from 0 °C to 70 °C.

For the current recirculation paths, the datasheet advises selecting fast-switching snubber diodes, preferably of Schottky type. A family of suitable devices is the SB2x0 [14], which are rated for a maximum average forward current of 2 A. As a first approximation, the supply voltage of the power output stage, plus one forward voltage drop of the snubber diode during clamping, bounds the maximum reverse voltage. As a consequence, the SB220 can be selected for the circuit.

The sense resistors must be sized to comply with the maximum allowable voltage at the sensing pins, equal to 2 V. Considering a $(680.0 \pm 6.8)\,\mathrm{m\Omega}$ resistor rated for 2 W [2], the nominal voltage drop for the chosen maximum phase current of 1.35 A is 918 mV. In this condition, the resistor dissipates 1.24 W: computing the thermal resistance from the power derating curve in the datasheet:

$$R_\theta = \frac{155\,^{\circ}\mathrm{C} - 70\,^{\circ}\mathrm{C}}{2\,\mathrm{W}} = 42.5\,^{\circ}\mathrm{C/W}$$

it is possible to evaluate the effect of self-heating. In the worst case of the 0 °C to 70 °C temperature range, the resistor heats to:

$$T_{\mathrm{r,max}} = T_{\mathrm{a,max}} + R_\theta\, P \approx 123\,^{\circ}\mathrm{C}$$

Consequently, from the temperature coefficient $\alpha = 300\,\mathrm{ppm/^{\circ}C}$:

$$\Delta R_{\mathrm{sh}} = R_0\, \alpha\, (T_{\mathrm{r,max}} - T_{\mathrm{r}}0) \approx 19.9\,\mathrm{m\Omega}$$

which shows that the self-heating effect dominates the uncertainty. Accordingly:

$$R_7 = R_8 = (680 \pm 27)\,\mathrm{m\Omega}$$

The sensed voltage is used in the L6506 to implement the constant current drive. In the worst case, the phase current reaches the maximum value of 1.35 A and the sensed voltage at the comparator input

is 882 mV. As a consequence, the major constraints are:

$$V_{\mathrm{ss}} \frac{R_4}{R_3 + R_4} \leq 882\,\mathrm{mV}$$
$$\frac{V_{\mathrm{ss}}}{R_3 + R_4} \gg I_{\mathrm{IB}}$$

Having already derated the maximum phase current, the constraint can be evaluated for the nominal resistance values. An optimal selection is:

$$R_3 = 22\,\mathrm{k\Omega} \qquad\qquad R_4 = 4.7\,\mathrm{k\Omega}$$

In fact, with 1 % uncertainties, the worst case voltage reference would be 895 mV, which corresponds to a maximum phase current of 1.37 A. In addition, the phase current may exceptionally increase above the chosen peak value during the active time of the oscillator waveform, as the driving inputs cannot be masked. With the same selection of (R_2, C_4) as in the application circuit [11, p. 5], the nominal frequency of the oscillator is 16.9 kHz and the nominal active time lasts 2.57 µs.

A PS/2 Controller

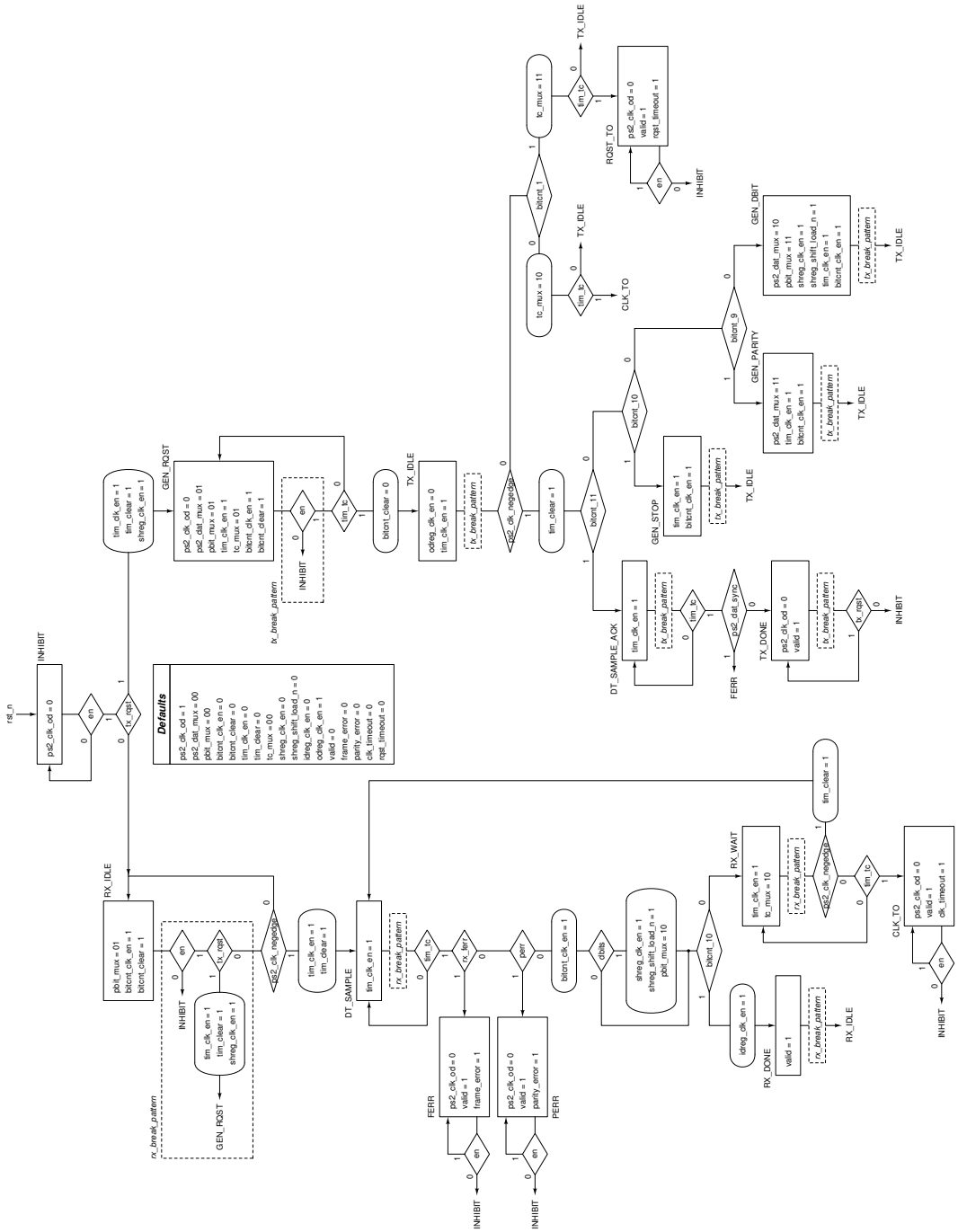


Figure 8: PS/2 controller detailed ASM chart

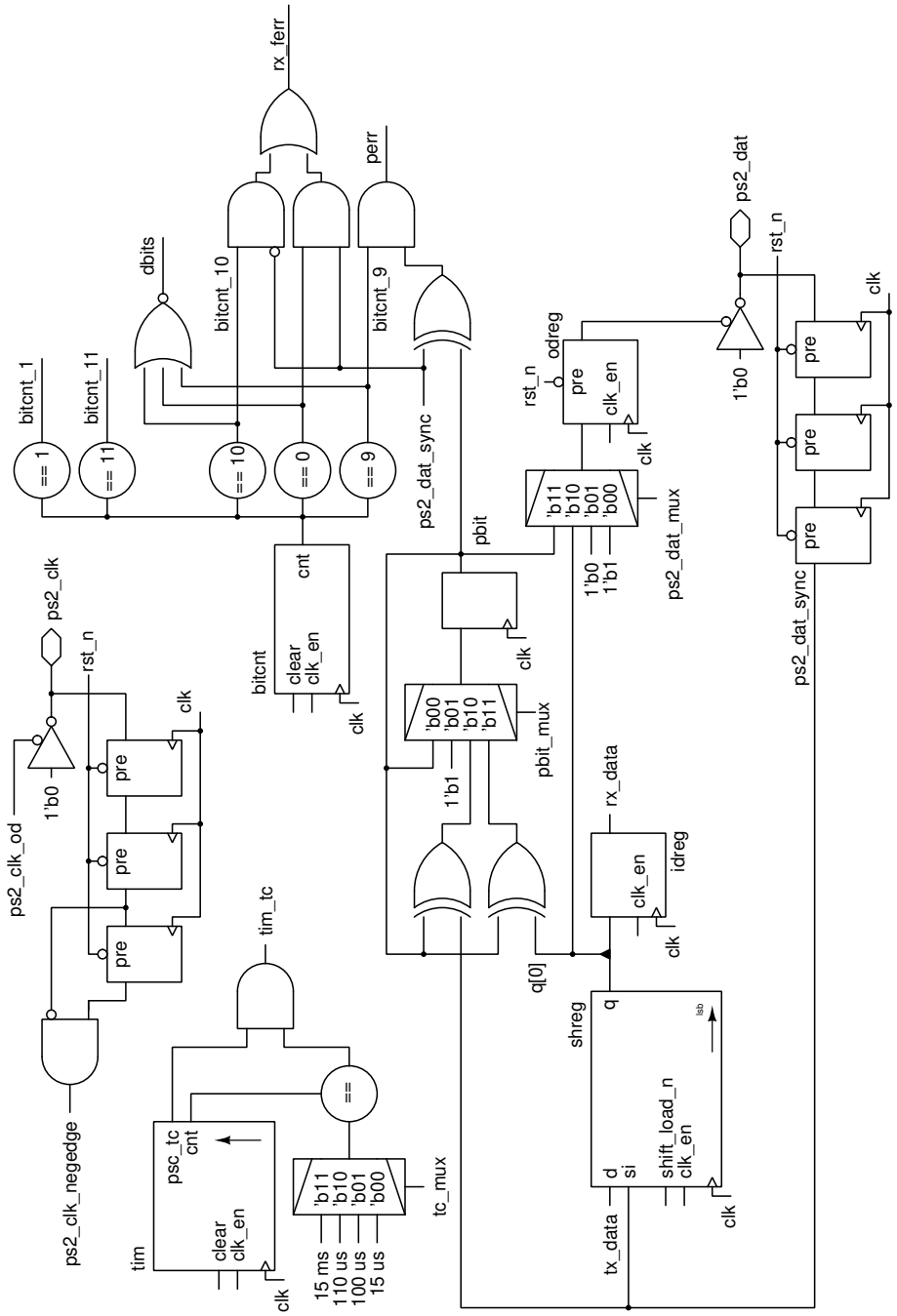


Figure 9: PS/2 controller data path

B Show-Ahead Single-Clock FIFO

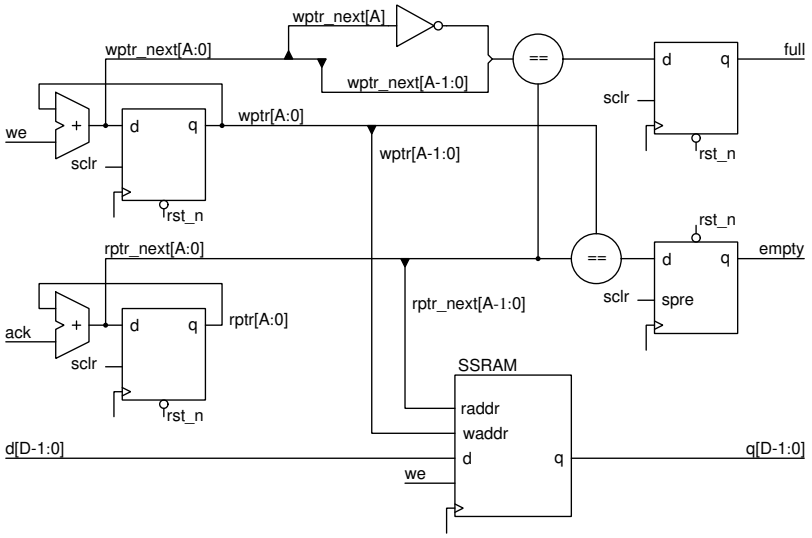


Figure 10: Show-ahead single-clock FIFO design

C SPI Slave Controller

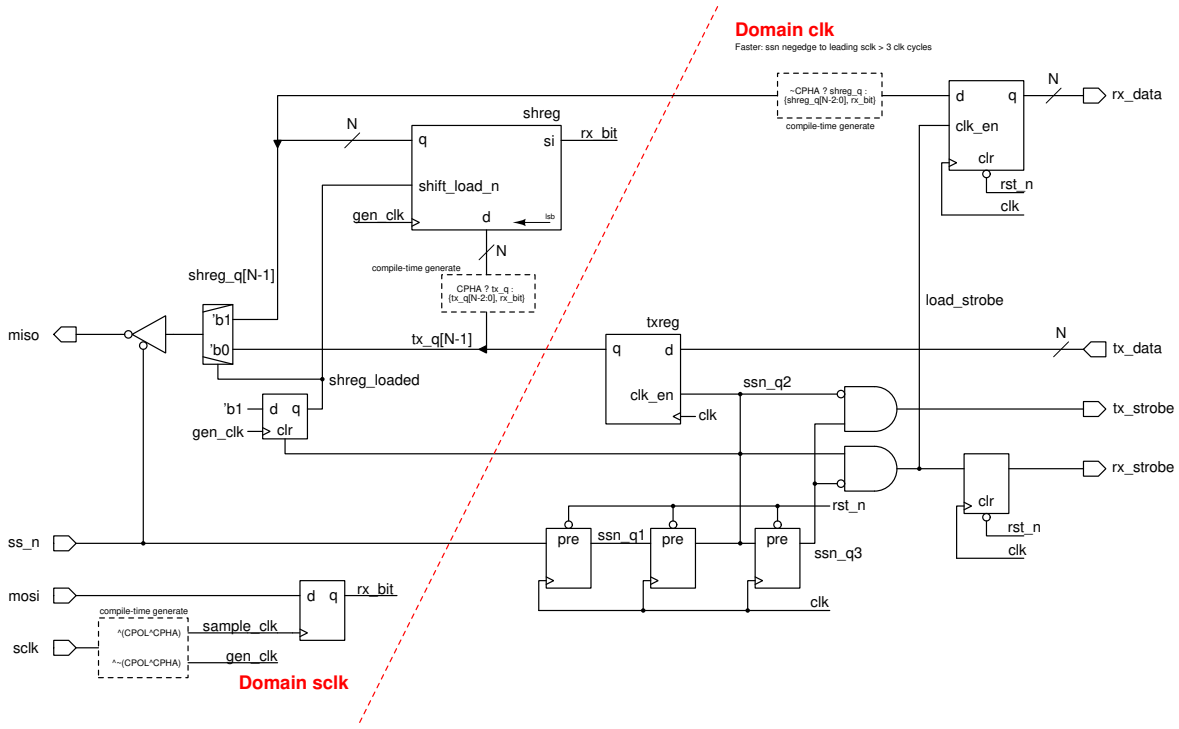
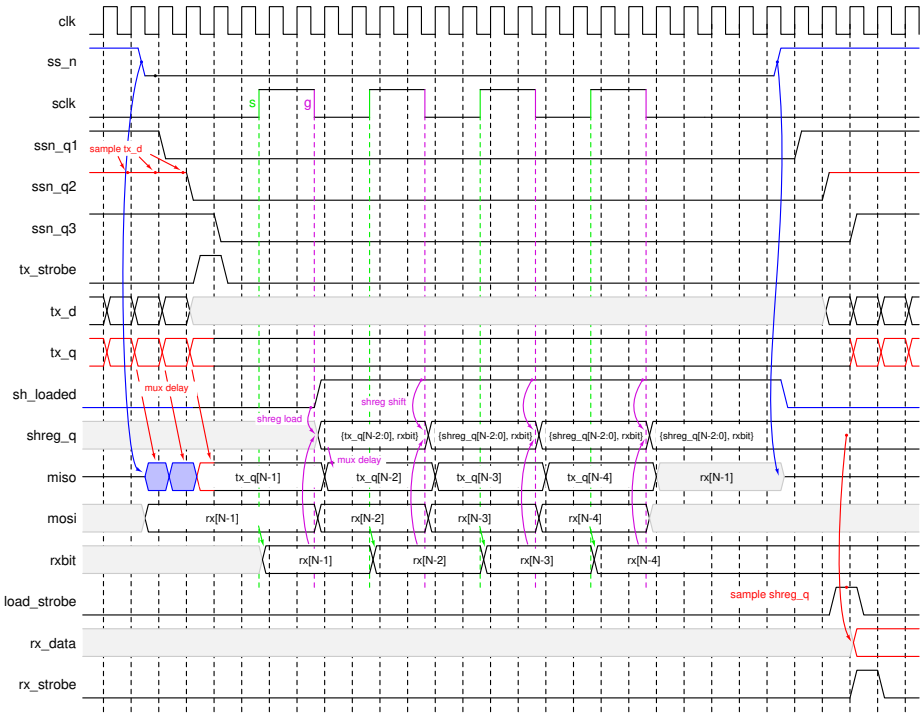
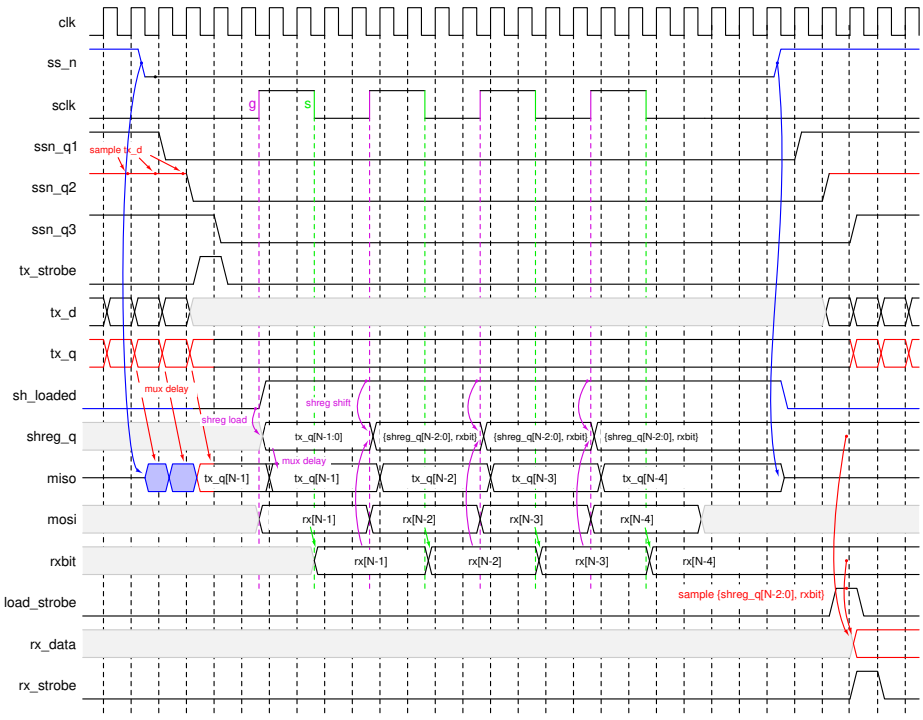


Figure 11: SPI slave controller



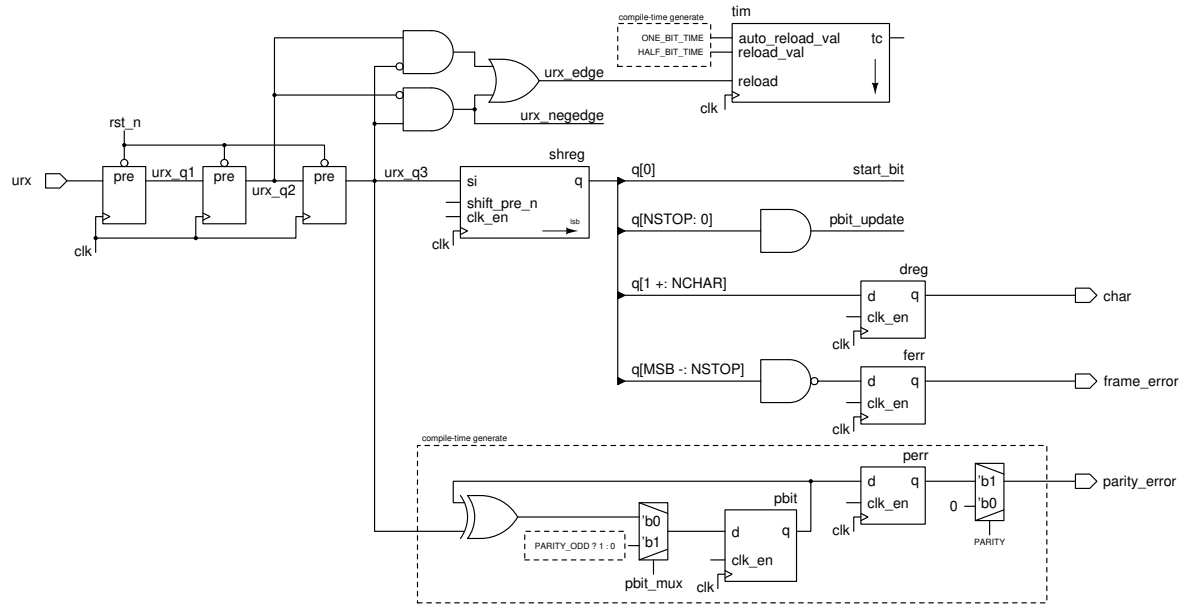
(a) 4 bit transmission, CPOL = 0, CPHA = 0



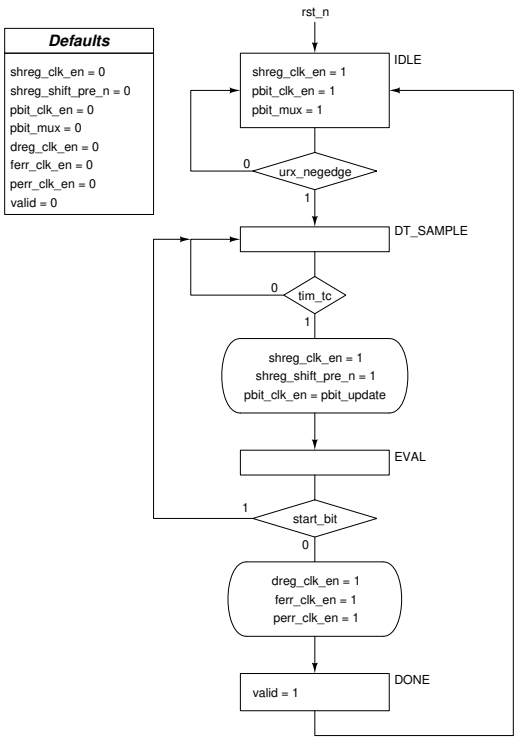
(b) 4 bit transmission, CPOL = 0, CPHA = 1

Figure 12: Manual timing analysis for the SPI slave controller

D UAR(T) Core



(a) Datapath



(b) ASM detailed chart

Figure 13: UART receiver-only core design