Happy Birthday Design Problem

Overview

The Happy Birthday Design Problem aims to implement a simple digital system comprising a Transmitter and Receiver. The system transmits a randomly generated bit stream and the receiver detects your date of birth in the sequence and displays the number of times it is detected on a seven segment display.

Encoding Format

Field	Bit Width	Description	Example
Month	4 bits	Encodes the month number (1–12)	July = 0111
Date	5 bits	Encodes the date (1–31)	10 = 01010
Total	9 bits	Combined sequence (Month + Date)	011101010

Architecture

The system contains two major blocks, the data generator & transmitter and the detector & BCD encoder. The system operates on a 10KHz clock source common for trans-receive blocks.

TRANSMITTER

The Transmitter Block generates 10-bit data words and transmits them serially at a defined rate using the system clock.

- Generator: A 10-bit counter acts as the data source, producing values from 0 to 1023. The counter increments once per completed transmission, ensuring that a new 10-bit word is generated only after the previous one has been sent. This prevents overlap between data generation and transmission. (Slow Decade Counter: HDL Bits)
- Each 10-bit value is transmitted one bit per clock cycle, starting with the least significant bit (LSB). The transmitter therefore outputs a continuous serial stream synchronized with the 10 kHz system clock.

Control Flow:

1. When i_tx_ena_n is low, transmission is active.
2. The transmitter shifts out 10 bits, one per clock, on its output line.
3. After 10 cycles, a tx_done pulse signals completion.
4. The generator counter increments once on tx_done, preparing the next 10-bit word.

Timing Summary:

- Clock frequency: 10 kHz
- Bits per frame: 10
- Bit rate: 10 kHz
- Frame rate: 1 kHz

Thus, each full 10-bit frame is sent every 1 ms, and the complete counter cycle repeats approximately every 1.024 s.

RECEIVER

The Receiver Block monitors the incoming serial data stream and detects a fixed 9-bit birthday pattern (Month + Date). Each successful detection increments a counter, whose value is displayed once per second.

- The receiver samples one bit per clock from the shared 10 kHz system clock. The incoming data is LSB-first, matching the transmitter's bit order. The receiver must maintain a running record of the last nine received bits for pattern comparison. The design approach is open-ended:
  - You may implement the detector as a finite state machine (FSM) that transitions through states according to partial pattern matches, or
  - Use a shift-register and comparator method that checks all nine bits in parallel each clock cycle.

  Either method must ensure correct handling of overlapping pattern occurrences, meaning the same bit sequence can contribute to multiple detections if it reappears with shared bits.
- Each valid detection increments a counter. The counter may be implemented as a simple up-counter with synchronous reset. It records the total number of matches observed during the current measurement interval.

Display Update (Bonus Feature):

To quantify detections, the receiver includes a mechanism to present the hit count on a seven-segment display once every second.

At each one-second interval (derived from a clock divider on the 10 kHz system clock):

- The current hit count value is latched and transferred to the display driver, encoded in BCD.
- The hit counter resets to begin counting afresh for the next second.
- A validity pulse (o_hit_count_valid) may be asserted to indicate new display data.

This ensures that the display represents the number of successful detections per second, synchronized with the system clock.

Parameter	Value	Description
Sampling Rate	10 kHz	One bit received each clock
Detection Rate	10 kHz	One comparison per clock
Display Update Rate	1 Hz	New value latched each second
Pattern Length	9 bits	Month[3:0] + Date[4:0]
Counter Width	Configurable	Chosen to avoid overflow within one second

MATHS BEHIND THE LOGIC

- The system runs on a 10 kHz master clock shared by both transmitter and receiver. Each data word has 10 bits, so the effective transmission rate is:

$$f_{TX} = f_{CLK} / 10 = 1 \text{ kHz}$$

- This means one complete 10-bit word is transmitted every 1 millisecond. The data generator is a 10-bit counter producing 1024 values (0–1023). At a 1 kHz transmission rate, it completes one full cycle in:

$$T_{cycle} = 1024 / 1 \text{ kHz} \approx 1.024 \text{ seconds}$$

– So the transmitted bit-stream repeats roughly once every second. For precise display updates, use a 1-second divider (10,000 clock cycles at 10 kHz) instead of relying on the 1.024-second generator rollover.

General relationships:

$$f_{TX} = f_{CLK} / N_{bits}$$

$$T_{cycle} = (2^{N_{counter}}) / f_{TX}$$

TOP LEVEL PORTLIST

Port Name	Direction	Width	Description
i_clk	Input	1 bit	System clock input operating at **10 kHz**.
i_rst	Input	1 bit	**Active-high synchronous reset** signal. Resets all counters and state elements when asserted.
i_tx_ena_n	Input	1 bit	**Active-low transmit enable**. Transmission logic is enabled when this signal is low (0).
o_hit_count	Output	[NumDig*7-1:0]	Seven-segment display control code representing the **hit count**. Each digit has 7 segment control lines. The number of digits (NumDig) is determined based on the maximum counter value to be displayed.
o_hit_count_valid	Output	1 bit	Indicates that o_hit_count holds a **valid hit count** value for display.

HOW SHOULD YOU PROCEED?

– Understand Requirements: Analyze functionality, timing, and display update logic.
– Plan Architecture: Create block diagrams and define module hierarchy.
– RTL Implementation: Develop and verify smaller blocks, then integrate them.
– Simulation: Run testbench to validate functionality and performance.

EVALUATION CRITERIA

This problem carries 5 marks with an extra bonus mark. You have a chance to score 6/5 in this assignment.

Criterion	**Description**	**Marks**
Understanding and Block Diagram Development	Clarity in understanding the system functionality and presenting it through a well-structured block diagram.	2
Implementation Status	Successful RTL implementation and integration of the required modules.	2
Code Quality	Clean, readable, and well-commented code following standard RTL practices.	1
Bonus: 1-Second Display Functionality	Implementation of the display update at 1-second intervals.	+1