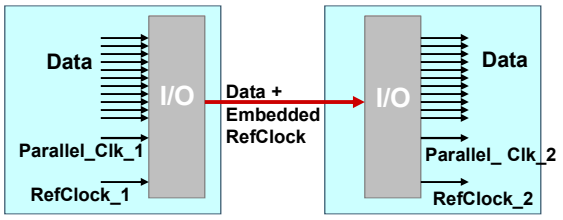
CAD for Digital Circuit Design
COMM 523

Course Project

Mohamed Dessouky
Integrated Circuits Laboratory
Ain Shams University
Cairo, Egypt
Mohamed.Dessouky@eng.asu.edu.eg
http://portal.eng.asu.edu.eg/mdessouky

High Speed Serial Links



Serial Link (SerDes)

- Send data serially instead of parallel busses.
- Example: USB, SATA, …
- High-speed clock (RefClock) = Serial transmission rate
- Parallel data clock (Parallel_Clk) = RefClock/Data bus width
- Characteristics
  - Point-to-point differential signaling → Higher line speeds
  - No separate clock signal (embedded clks), clock is extracted at the receiver.
  - Can use multiple parallel lanes → Bandwidth scalability

Packets (Frames)

- SerDes requires a method for aligning the incoming stream into words.
- Use Packets: a well-defined collection of bytes consisting of a header, data and a trailer.
- Size is chosen for enough SF symbols to align to and idle symbols to provide for clock correction.
  - The bigger, the less overhead and the more bandwidth available for data.
  - The smaller, the more alignment and clock correction characters.
- Data Encoding: Modifies data for enough transitions to facilitate clock recovery

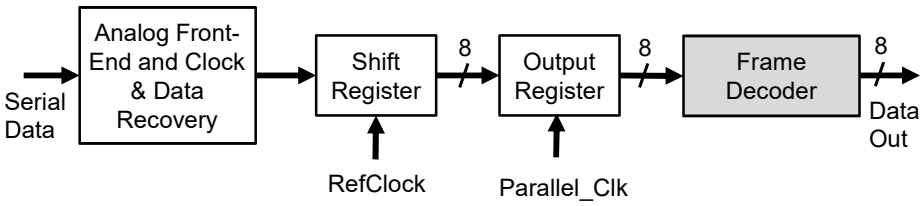
Needs a Communication Protocol

- Procedure to initialize the link and be sure that both sides are ready to receive data
- Line encoding algorithm
- Alignment, clock correction, idle sequences
- Packet composition
- Error correction algorithm

- Standard *versus* Custom protocols??

Project: Frame Decoder

- At the Deserializer side.
- After serial data reception, data is stuffed in an 8-bit shift register (serial-to-parallel).
- The output 8 bits are stored in an output register.
- The "Frame Decoder" extracts the data from the sent frame.



Deserializer block diagram

Frame Definition

SOF	ID	Address	Data	EOF
7E	80	1 Byte	1 Byte	E7

Simple Data Frame

SOF	ID	Address	Data	EOF
7E	81	1 Byte	MSB LSB	E7

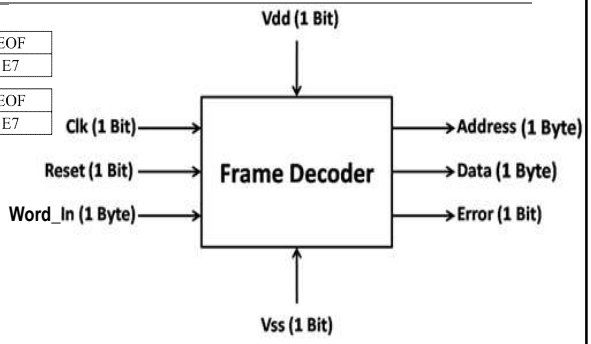
Extended Data Frame

- The Communication Protocol uses 2 different Data Frames
- Each Frame begins with a Start of Frame (SOF) followed by the frame ID: 80 for Simple Frame and 81 for the Extended Frame.
- Then, both frames provide the frame address: 1 Byte for both.
- Next, the data string which is 1 Byte for the Simple Frame and 2 Bytes for the Extended Frame.
- Finally, an End of Frame (EOF) string marks the end of the received frame.

Frame Decoder: Inputs

SOF	ID	Address	Data	EOF
7E	80	1 Byte	1 Byte	E7

SOF	ID	Address	Data	EOF
7E	81	1 Byte	MSB LSB	E7

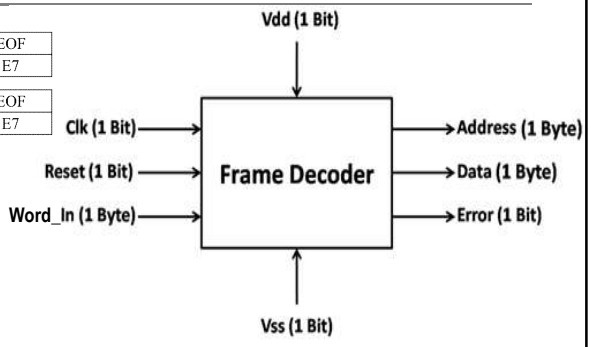


- Clk: Input Parallel_Clk clock signal at 10MHz.
- Word_In: Main data input. At each clock rising edge, 1 Byte is received.
- Reset: Reset to an initial state waiting for a SOF at Word_In.
- Vdd/Vss: Supply and Ground.

Frame Decoder: Outputs

SOF	ID	Address	Data	EOF
7E	80	1 Byte	1 Byte	E7

SOF	ID	Address	Data	EOF
7E	81	1 Byte	MSB LSB	E7



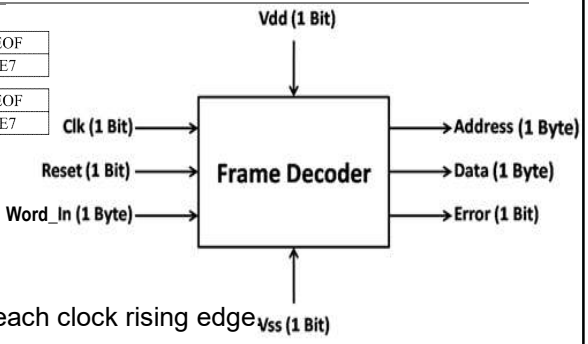
- Address: Address of the received frame
- Data: Data output, either 1 Byte or two consecutive Bytes in series.
- Error: Error bit, that is normally equal to '0'. It is asserted (set to '1') if the received frames don't follow the given structure, i.e. if the ID is NOT '80' nor '81', or after the data there is no EOF=E7.

Note: Initialize all outputs to zero after a Reset.

Frame Decoder: Operation

SOF	ID	Address	Data	EOF
7E	80	1 Byte	1 Byte	E7

SOF	ID	Address	Data	EOF
7E	81	1 Byte	MSB LSB	E7



- Word_In receives 1 Byte each clock rising edge
- If the received byte is a SOF, then check the following Byte:
  - If it is not 80 or 81, output Error and wait for a new SOF.
  - If it is 80, then the following **two** Bytes are Address and Data
  - If it is 81, then the following **three** Bytes are Address, MSB and LSB Data
- If after receiving the data, the Word_In
  - is not E7, output Error and wait for a new SOF.
  - is E7, just wait for a new SOF.

Project Implementation

- Project statement and deadline, see course web site.
- Guideline files throughout the lectures. Check the "Course Project Files" folder.

First Step – Project Part 1:

- Design the state diagram. Choose Mealy or Moore outputs. Must explicitly state your choice in the documentation.
- Implement the FSM in VHDL.
- Prepare a ModelSim testbench to validate your design with proper assertions to be used throughout the project.
  - The more the assertions, the more effective the testbench will be in testing different phases of the design.

- Best to do each part of the project after the lecture directly and be prepared for the next step.