

Asynchronous FIFO & UART

Professor Yong Ho Song

■ First-In, First-Out

- + Queue의 동작과 유사
- + Write 된 순서대로 read 됨
 - Write & read pointer로 관리

■ Clock 기반의 module

- + Read 부분의 clock domain과 write 부분의 clock domain이 다름 (asynchronous)
- + 서로 다른 clock domain을 사용하는 device 들 간의 data 전달 통로

■ 주요 구성 modules

- + FIFO RAM : 실제 data를 저장
- + Pointer modules
 - Read pointer : read pointer 관리, empty 여부 판별
 - Write pointer : write pointer 관리, full 여부 판별
 - 두 pointer module은 서로 다른 clock domain 사용
- + Synchronization : read & write pointer 간 synchronization

Block diagram of Asynchronous FIFO

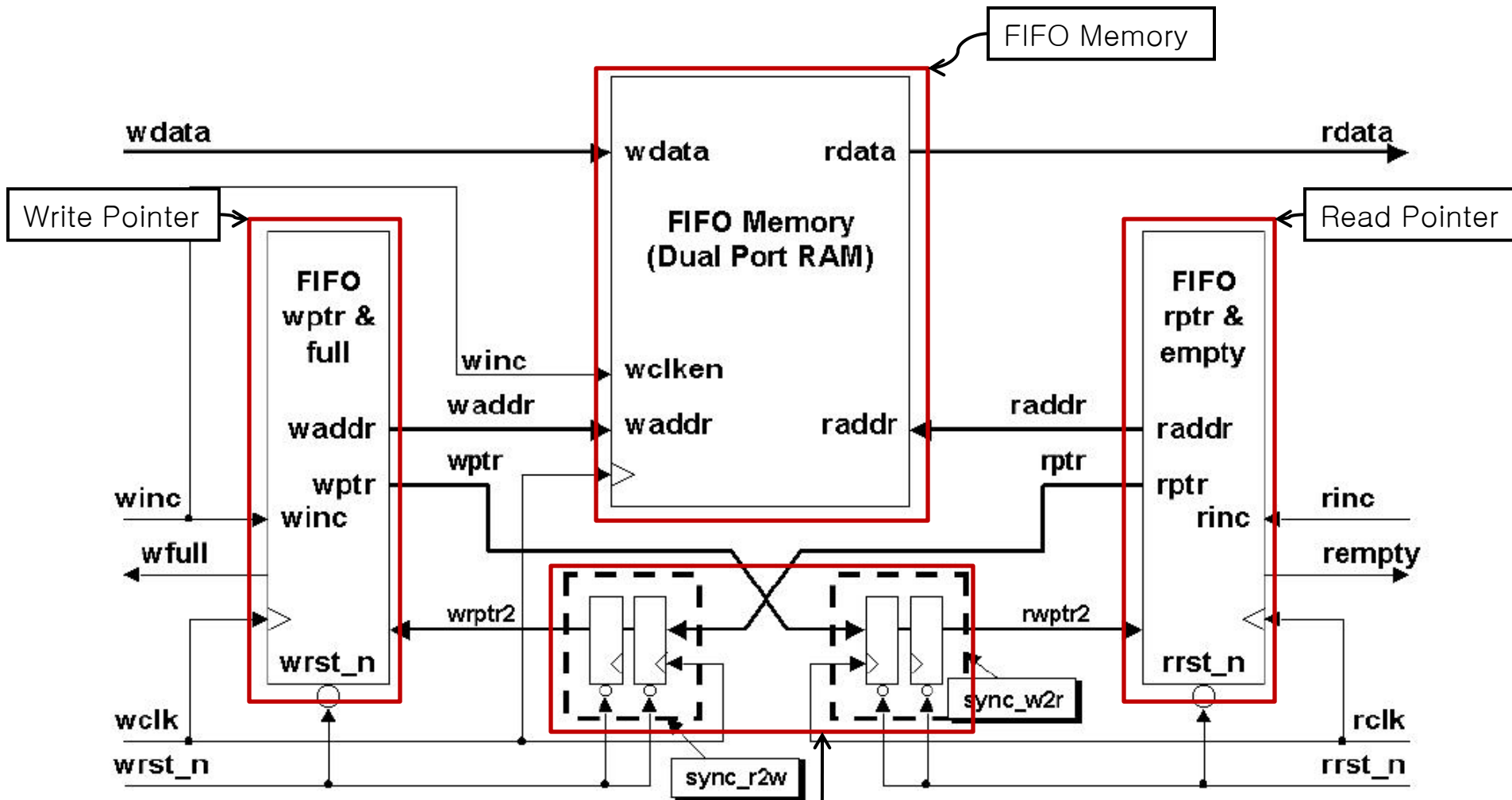


Figure 4 - FIFO1 partitioning with synchronized pointer comparison

Write - Read pointer간
Synchronize 역할

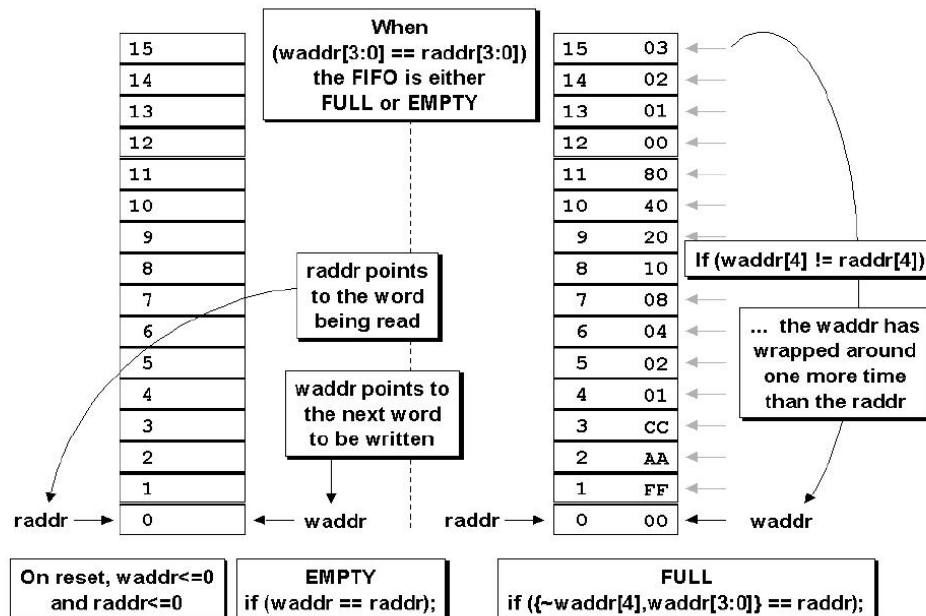
Read & write pointers

정의

- + Read pointer : 현재 읽어낼 부분을 가리킴
- + Write pointer : 현재 쓰여질 부분을 가리킴

Full & empty conditions

- + Empty : read pointer와 write pointer가 같은 부분을 가리킬 때
- + Full : empty의 조건과 동일
 - 구분 : 하나의 여유 bit를 두어 write가 한 바퀴 일어나면 set, 구분





■ 필요성 : Full & empty 여부 판별시

- + Read pointer module : Empty 여부를 판별하기 위해 write pointer 정보가 필요
- + Write pointer module : Full 여부를 판별하기 위해 read pointer 정보가 필요
- + 두 module은 서로 다른 clock domain을 사용 : sync. 가 맞지 않음
 - 서로의 data를 받기 위해서는 synchronization 필요

■ 해결책 : Buffer 구현

- + Read pointer module
 - Read 쪽 clock domain 사용
 - Write pointer 정보를 저장 후, read clock domain에 의해 data 전달
- + Write pointer module
 - Write 쪽 clock domain 사용
 - Read pointer 정보를 저장 후, write clock domain에 의해 data 전달



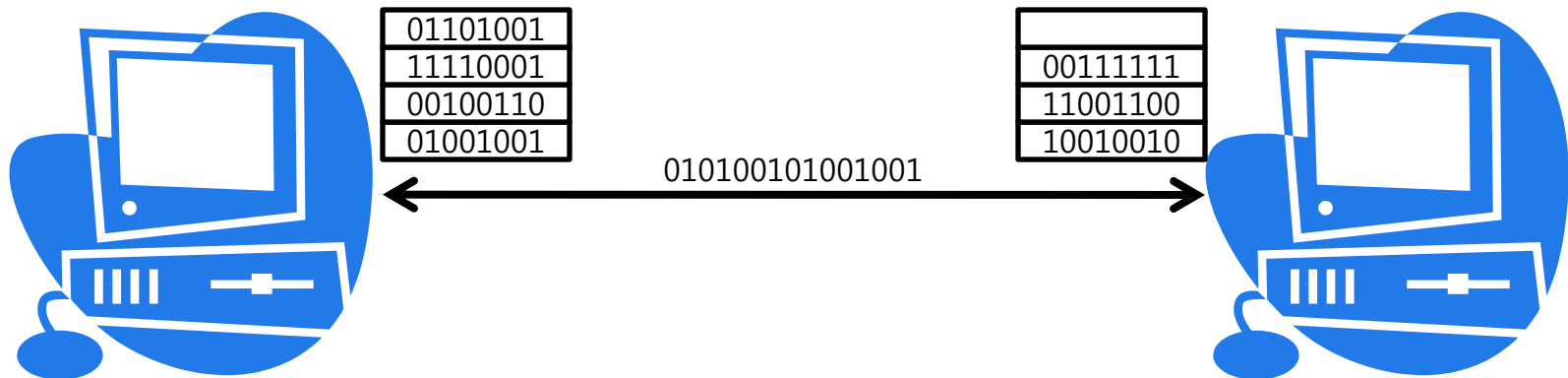
■ Size

- + Data size : 4 bits
- + FIFO RAM size : 16 Bytes (4 bits * 32개)
- + Parameter화 하여 변경하기 쉽도록 할 것 (UART part에 대비)

■ 그 이외

- + Write & read pointers, synchronization 부분 : 배운 대로 구현
- + Clock domain
 - Read : Spartan board oscillator를 그대로 활용
 - Write : Spartan board oscillator의 1/2로 활용 (저번 주 수업 내용 참조)
- + I/O
 - Inputs : switch 사용
 - Outputs : LED 중 4개를 택일하여 사용
 - Input / output triggers : Push buttons를 선택하여 사용
- + 나머지 부분 : 자유롭게 구현
 - Internet 검색 활용할 것 (단, source를 그대로 가져오는 행위는 불허)

■ RS-232 규약의 Serial 통신

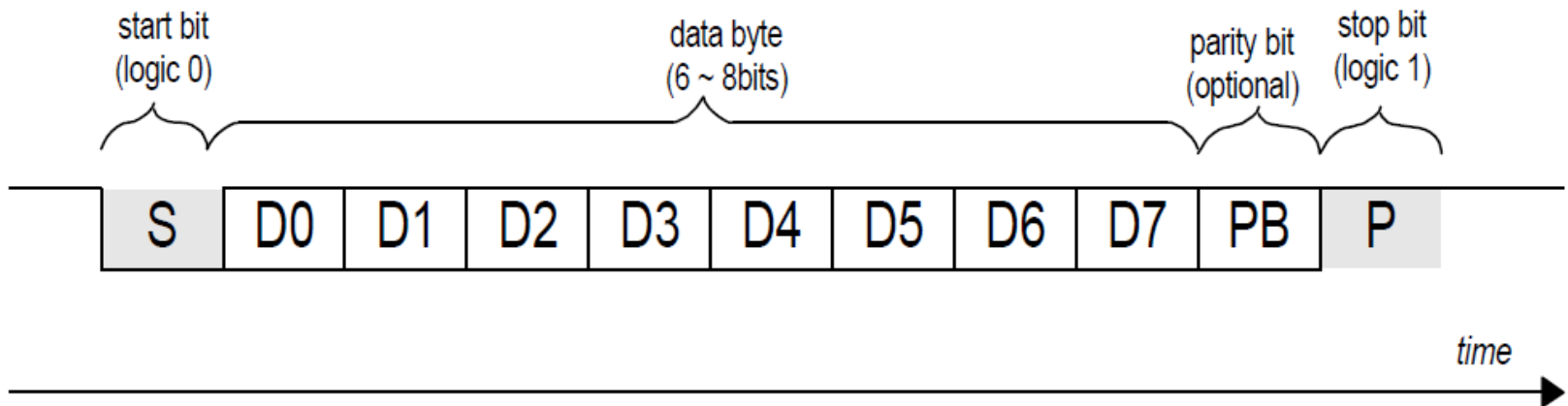


■ Simple Protocol

- + 1 Start bit, 1Parity and 1 Stop bit
- + Various Data Length : 6, 7, 8

■ UART Data Packet

- + 수신측과 송신측의 Baud rate가 미리 약속되어야 함
 - 별도의 Sync Signal(clk)을 사용하지 않음
 - Uart 내부 모듈은 Sampling을 위해 Baud rate의 16배 빠르기의 clock을 사용
- + 모든 전송은 start bit로 시작
 - 1에서 0으로 변하는 시점을 Start로 인식
 - 처음 Data Bit는 Start 신호 후 24번째 Cycle에 샘플링
- + Data Bit는 매 16 Cycle마다 1 Bit 씩 샘플링



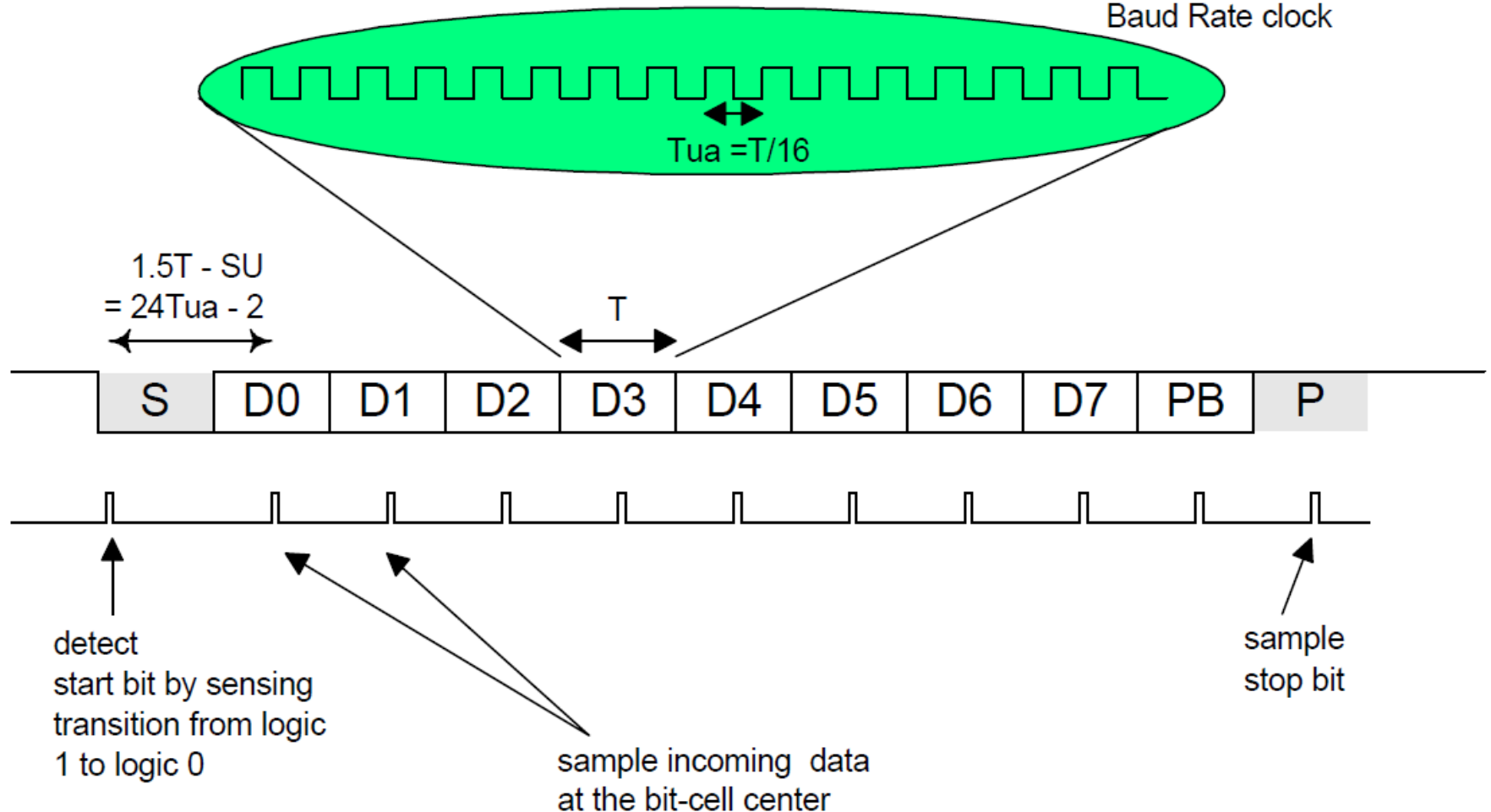
UART 동작 원리

$T = 1/\text{Baud Rate}$. Bit-cell period.

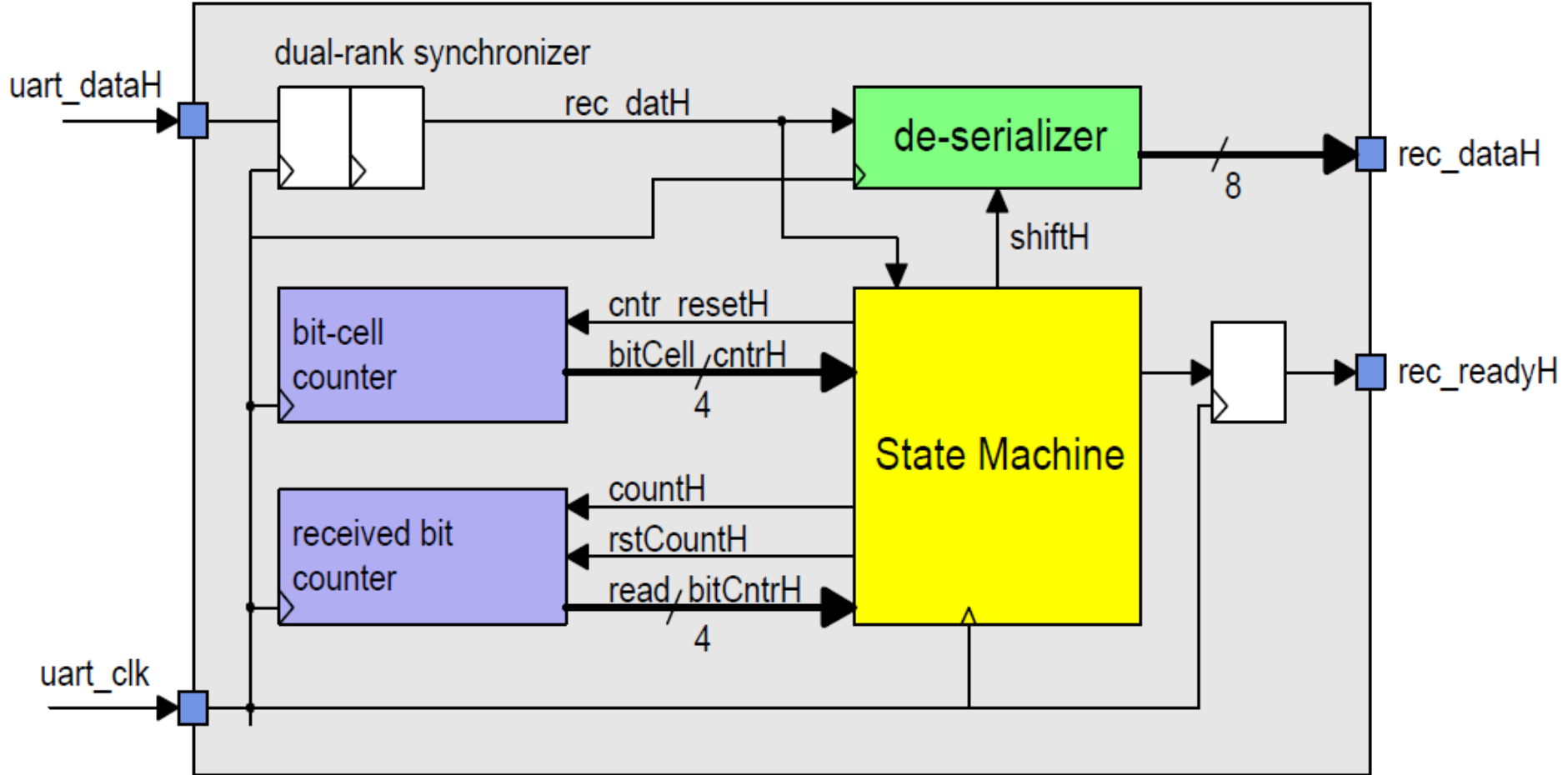
$T_{ua} = T/16$. Oversampled bit-cell period.

$SU = 2 T_{ua}$. Worst case synchronizer uncertainty.

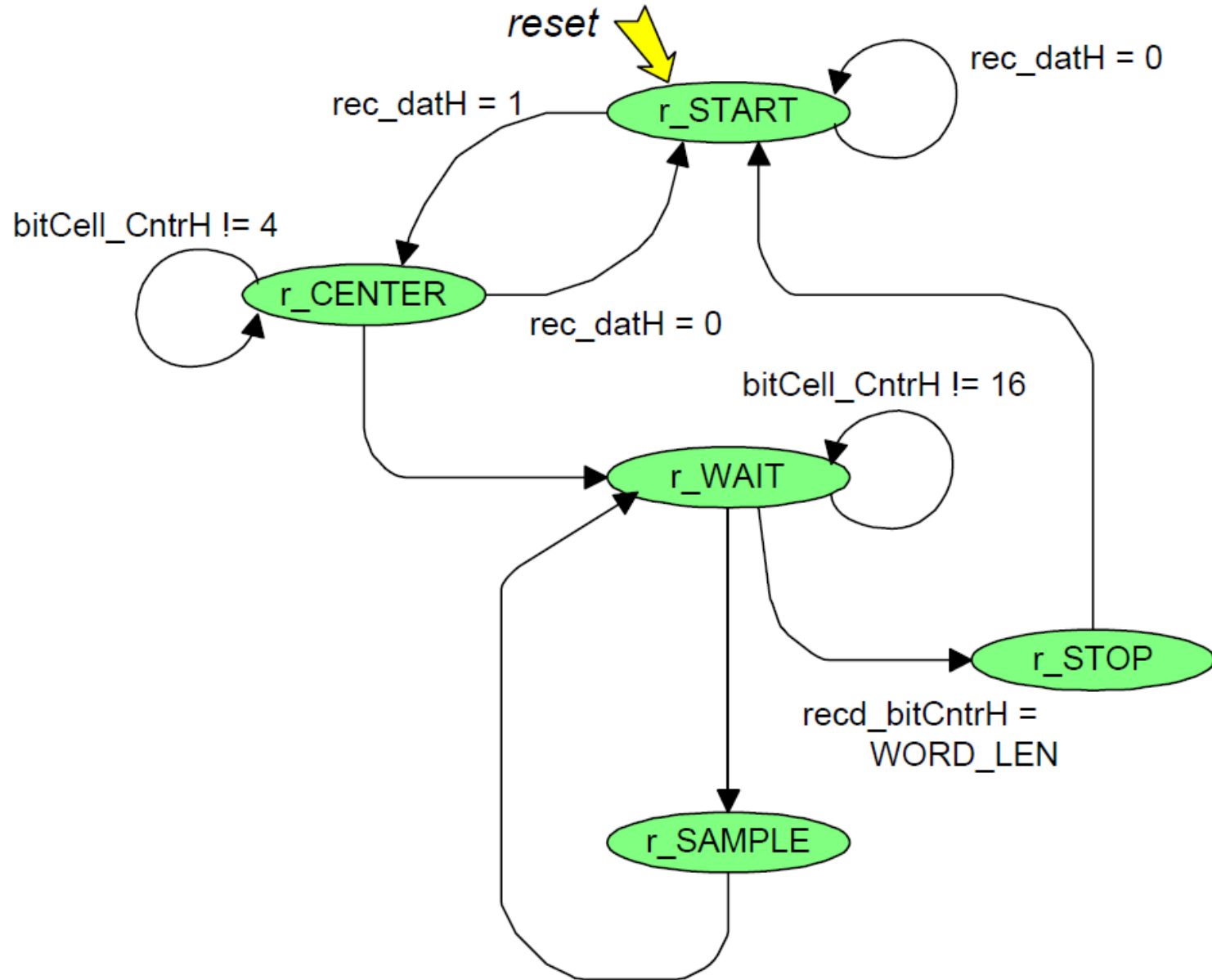
UART receiver samples the incoming data using $\times 16$ Baud Rate clock



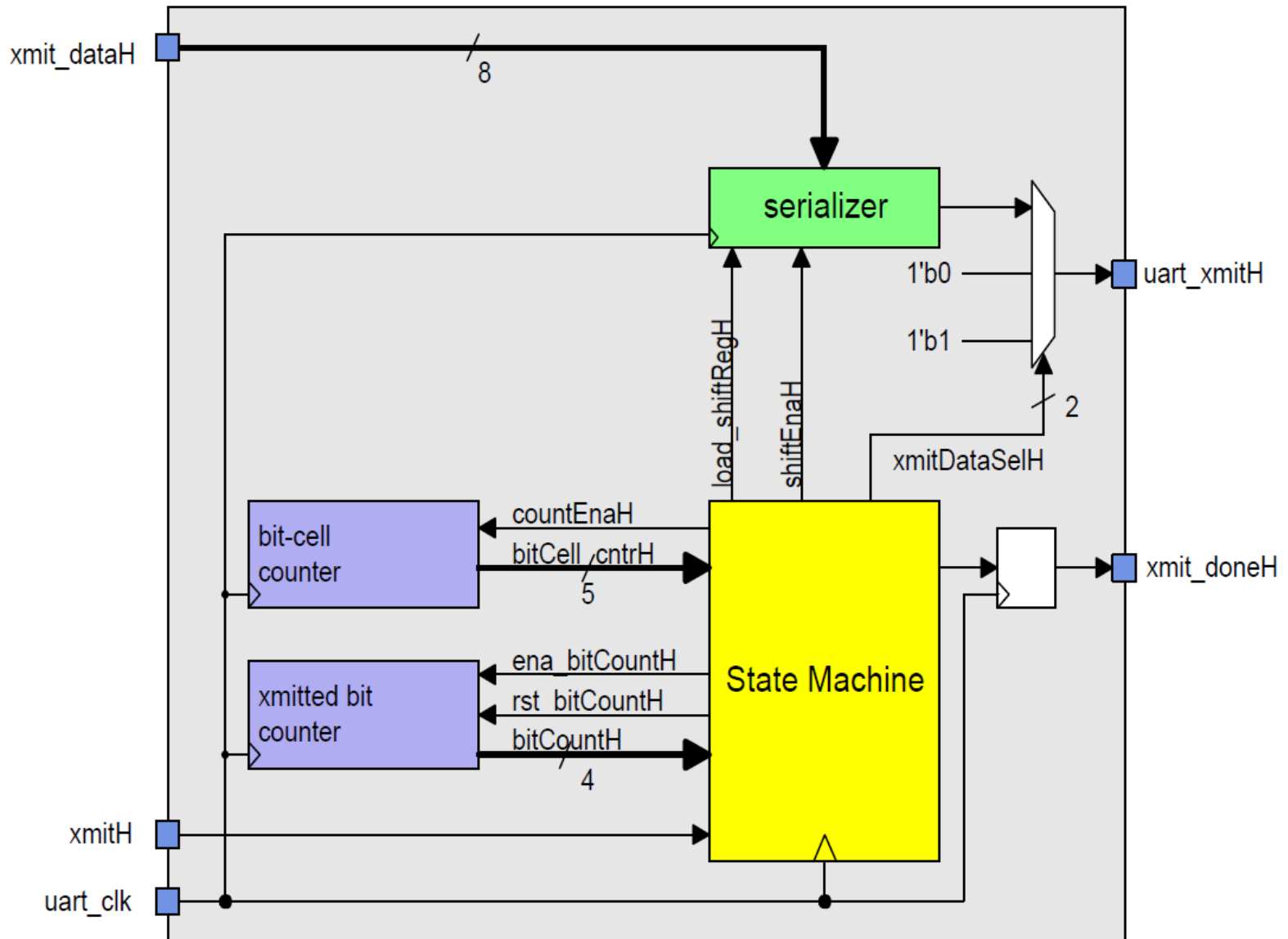
Functional Description of Receiver



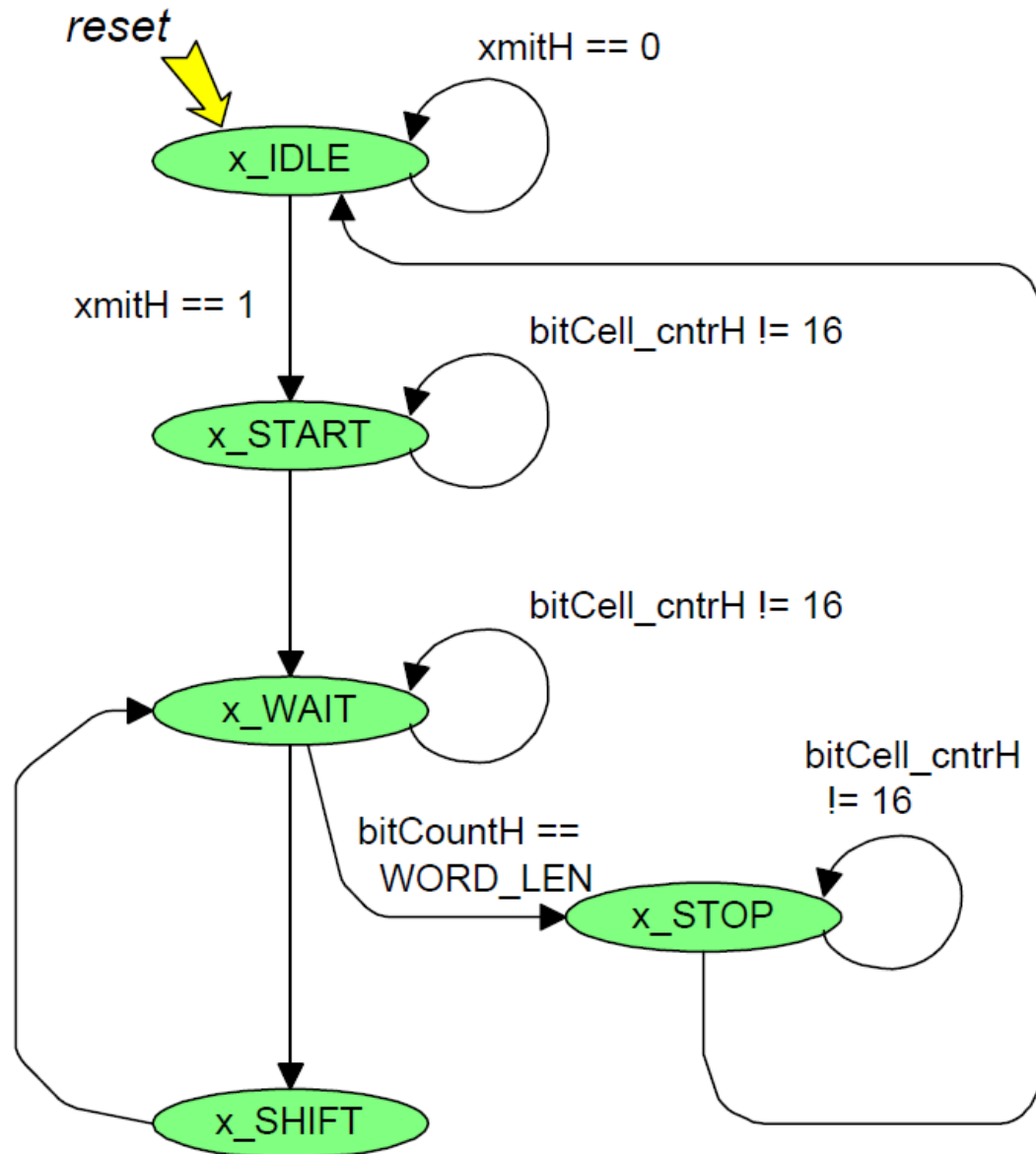
UART Receiver State Flow



Functional Description of Transmitter

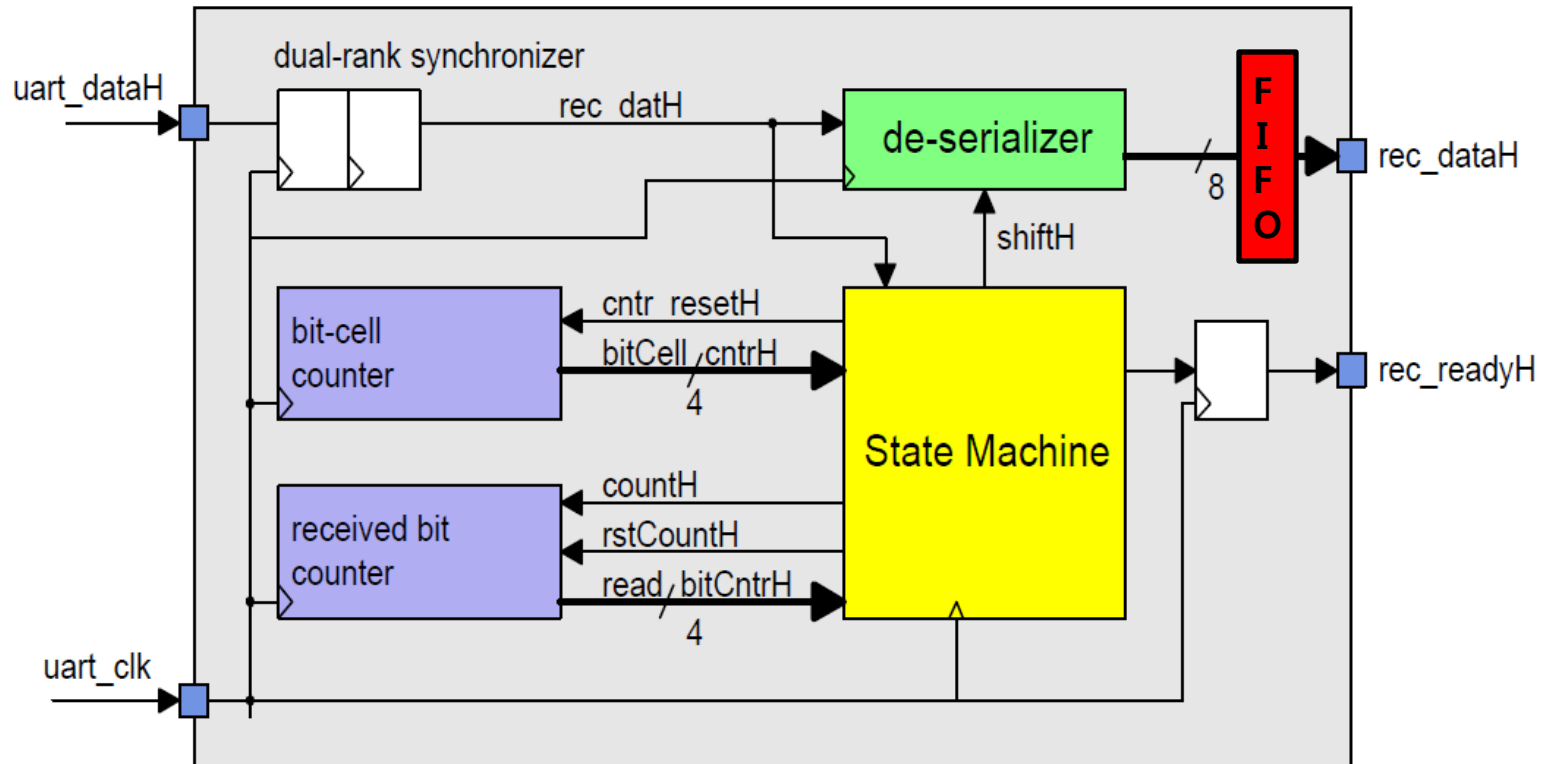


UART Transmitter State Flow



FIFO를 이용한 UART Receiver

- + UART의 Receiver 부분에 수신된 Data를 FIFO를 이용하여 출력



- + UART 송신은 PC 이용
- + Push Button을 누르면 LED에 데이터 표시



■ 제출 기한

+ 수요일반

- Soft copy : 5/26 (화) 자정까지 제출
- Hard copy : 5/27 (수) 수업 시간 시 제출

+ 목요일반

- Soft copy : 5/27 (수) 자정까지 제출
- Hard copy : 5/28 (목) 수업 시간 시 제출

■ 과제 수행을 위한 실습실 추가 사용 안내

+ Simulation waveform을 담당 조교에게 확인받은 후, 통과한 사람들에 한해서 실습실 및 Board 사용을 허가한다.

+ 반드시 해당 조 전원이 와서 확인을 받고, 학생증을 맡긴 후 사용하도록 한다.

- 확인을 받지 않은 상태에서 임의로 Board를 사용하다 적발 시 학점 1등급 감점 조치

+ 의문사항이 생길 경우 e-mail로 질문하기 바랍니다.

Thank You

Q & A