# RTL 설계교육

## (3주차)

2022.02.11

CoAsia SEMI Ltd.

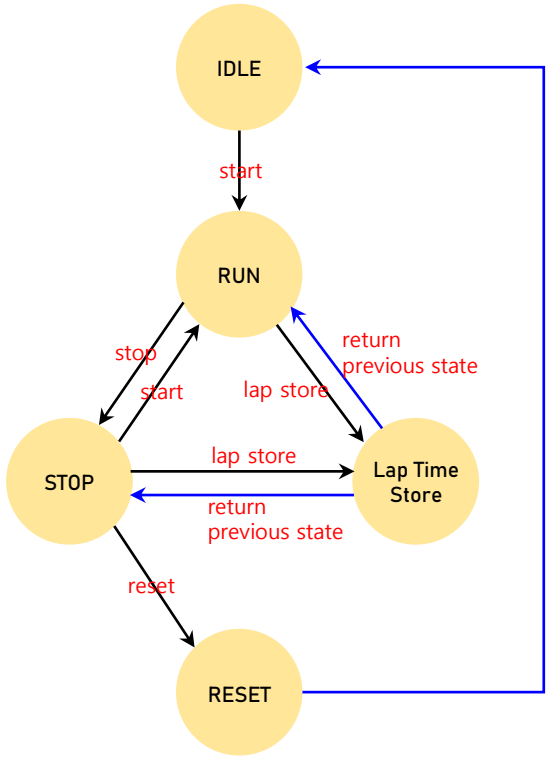
# Index

# 1. Stopwatch Review

## Specification

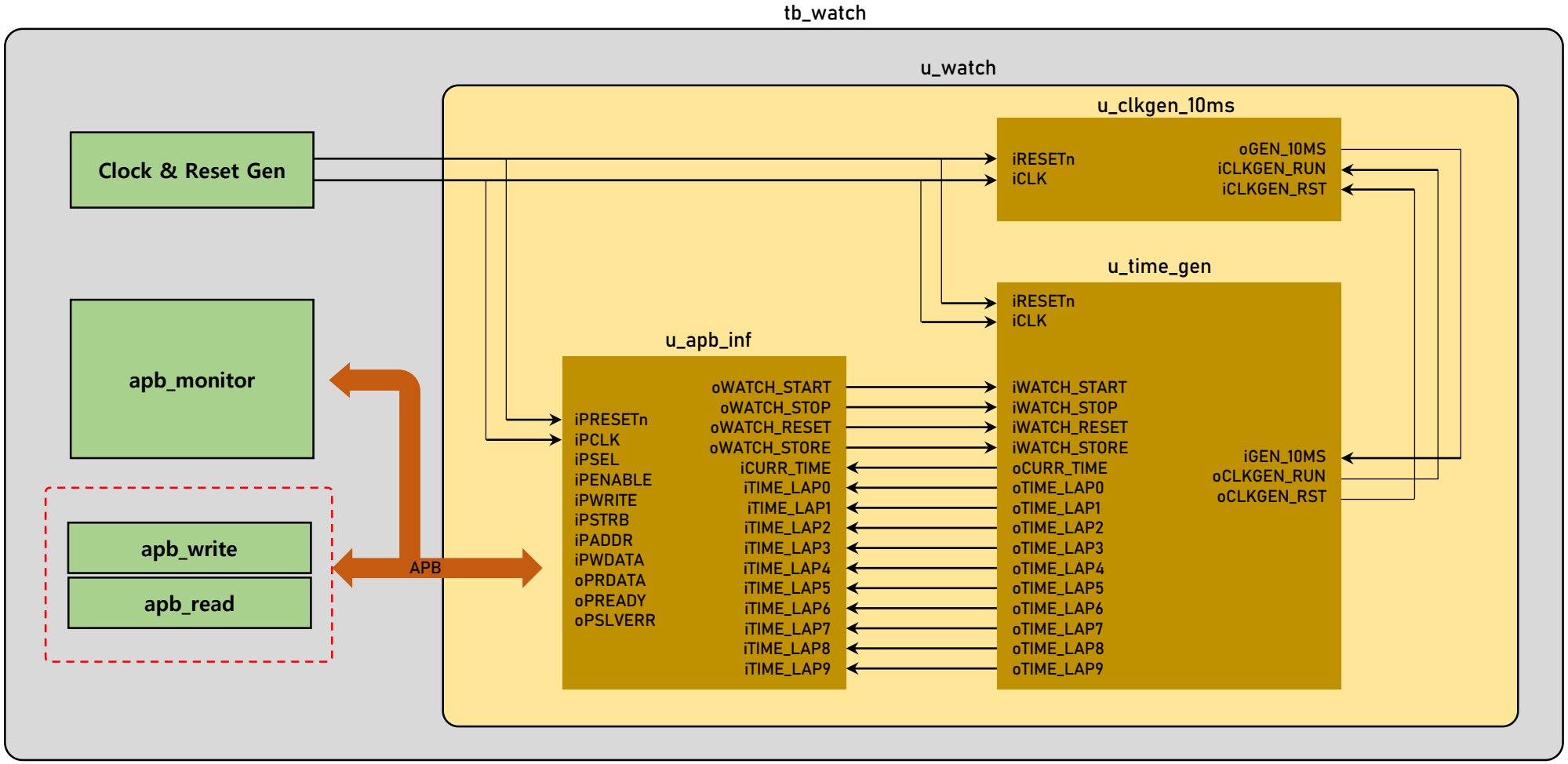
- 1/100초 분해능
- Start / Stop / Reset
- Lap Time (10개 지원)

- State Machine



# 1. Stopwatch Review

## Stopwatch Block Diagram



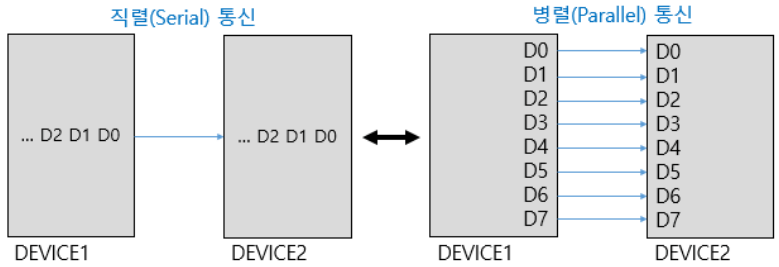
# 1. Stopwatch Review

## Verilog Code Review

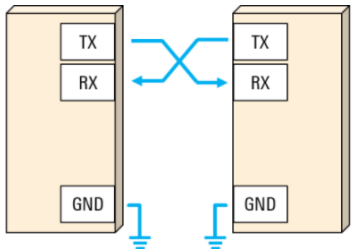
# 2. UART 란?

## UART

- 직렬(Serial) 통신 : 1개의 입출력 핀을 통해 n개 비트를 한 번에 전송하는 방법
- 병렬(Parallel) 통신 : n비트의 데이터를 전송하기 위해 n개의 입출력 핀을 사용하는 방법




- 범용 비동기 송수신기 (Universal Asynchronous Receiver / Transmitter)
- 송신기와 수신기가 공통의 클럭 신호를 공유하지 않음 (비동기 통신)
  - 송수신 양 방향에 동일한 비트 타이밍을 보장하기 위해 사전에 정의된 속도로 통신
  - Baud Rate (4800, 9600, 19.2k, 115.2k ..... )
  - 양방향에 동일한 프레임 구조 및 매개변수 사용



# 2. UART 란?

## UART 프레임

- Start Bit
  - Idle 상태에서 통신의 시작을 알림
  - High 상태에서 Low 상태로 바뀜
- Data Bit
  - 전송하고자 하는 User Data
  - LSB 먼저 전송
  - 5-9 bit 사이로 전송 가능 (주로 7bit or 8bit 사용)
- Parity Bit
  - Even Parity : 프레임에 포함된 "1"의 개수가 짝수가 되도록 구성
  - Odd Parity : 프레임에 포함된 "1"의 개수가 홀수가 되도록 구성
- Stop Bit
  - High 상태로 바뀜
  - 1bit or 2bit으로 변경가능

- 전송 예제 ("S" 전송)
  - Data Bit : 7
  - Parity Bit : Even Parity
  - Stop Bit : 1

❖ "S" → 0x53 (ASCII)

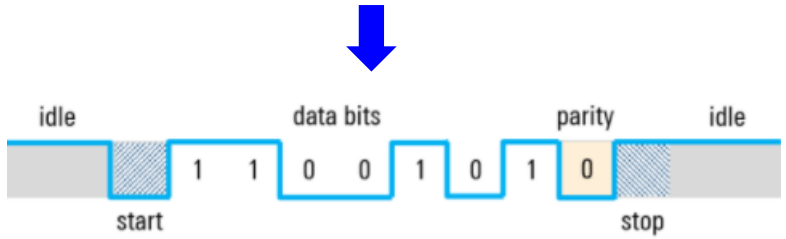
❖ 7'h53 → 2'b101_0011

❖ LSB First

101_0011 → 1100_101

❖ Parity

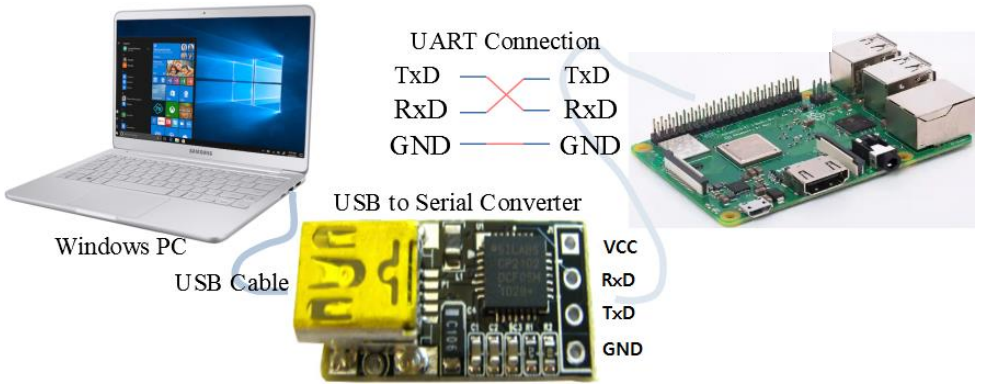
전체 "1"의 개수가 4개임으로 Parity Bit는 "0"



# 3. UART 설계

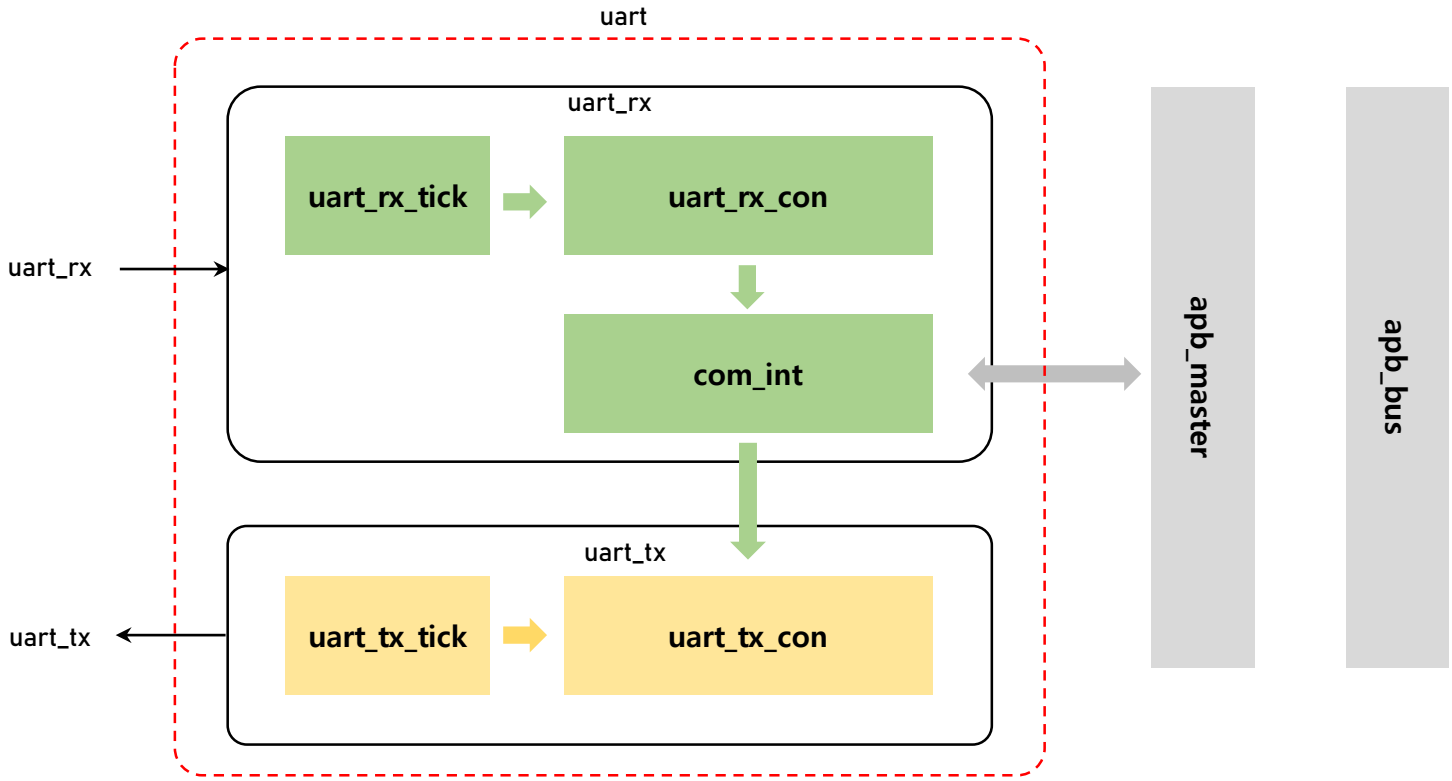
## Specification

- 프레임 구성
  - Data Bit : 8
  - Parity Bit : None
  - Stop Bit : 1
- Baud Rate
  - Default : 115200
  - 변경 가능해야 함
- Command Format
  - [COMMAND(W,R)] [ADDRESS(16)] [DATA(32)]
  - Address Range
    1. APB : 0x0000 – 0x0FFF
    2. UART Setting : 0xF000 – 0xFFFF
  - 예제
    1. W 1234 aa55aa55aa55
       - ✓ 1234번지에 aa55aa55aa55를 Write
    2. R 1234
       - ✓ 1234번지를 Read



# 3. UART 설계

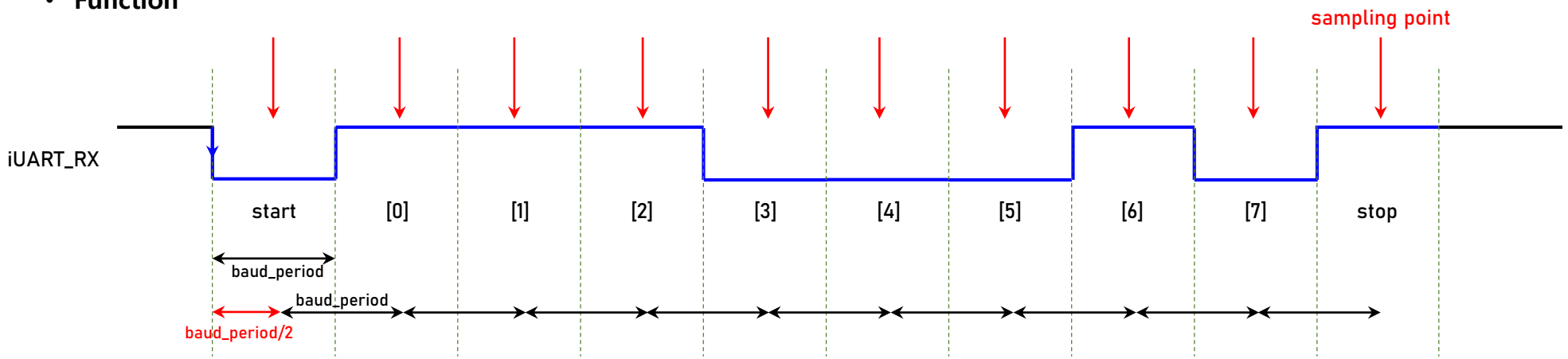
## Block Diagram



# 4. Module 설계 (uart_rx)

## uart_rx_tick

- **Function**




baudrate period count value = round ( 1 / baud_rate / (clock frequency)$^{-1}$ )

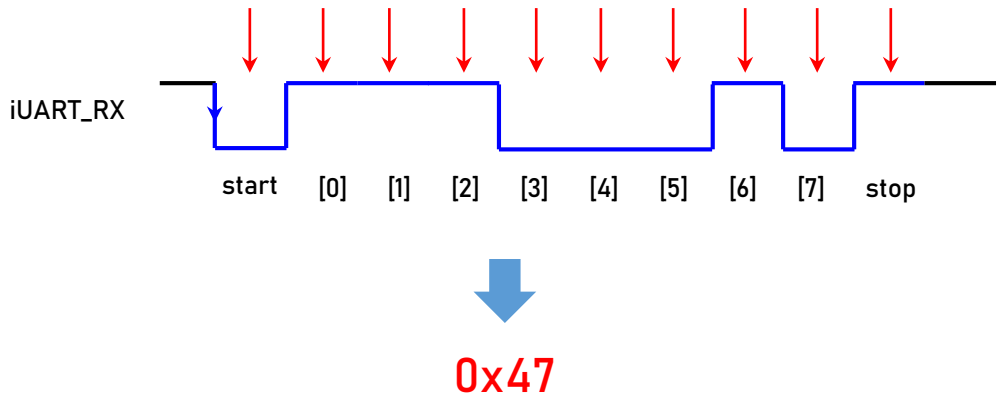
# 4. Module 설계 (uart_rx)

## Verilog Code

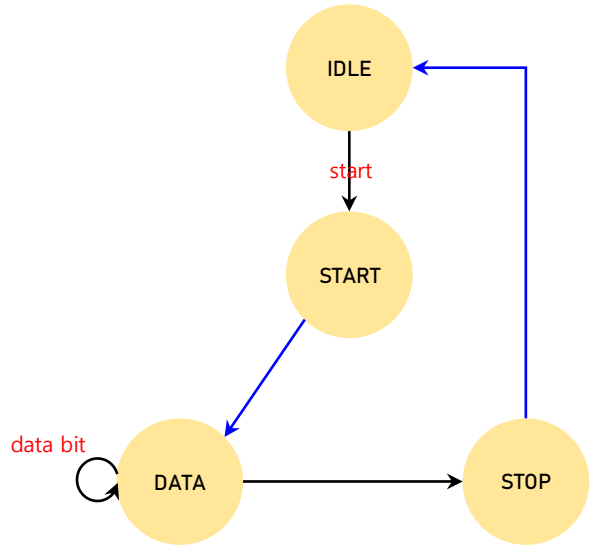
# 4. Module 설계 (uart_rx)

## uart_rx_con

- **Function**
  - ➢ **Serial Data를 Parallel로 변환**




- **State Machine**



# 4. Module 설계 (uart_rx)

## Verilog Code

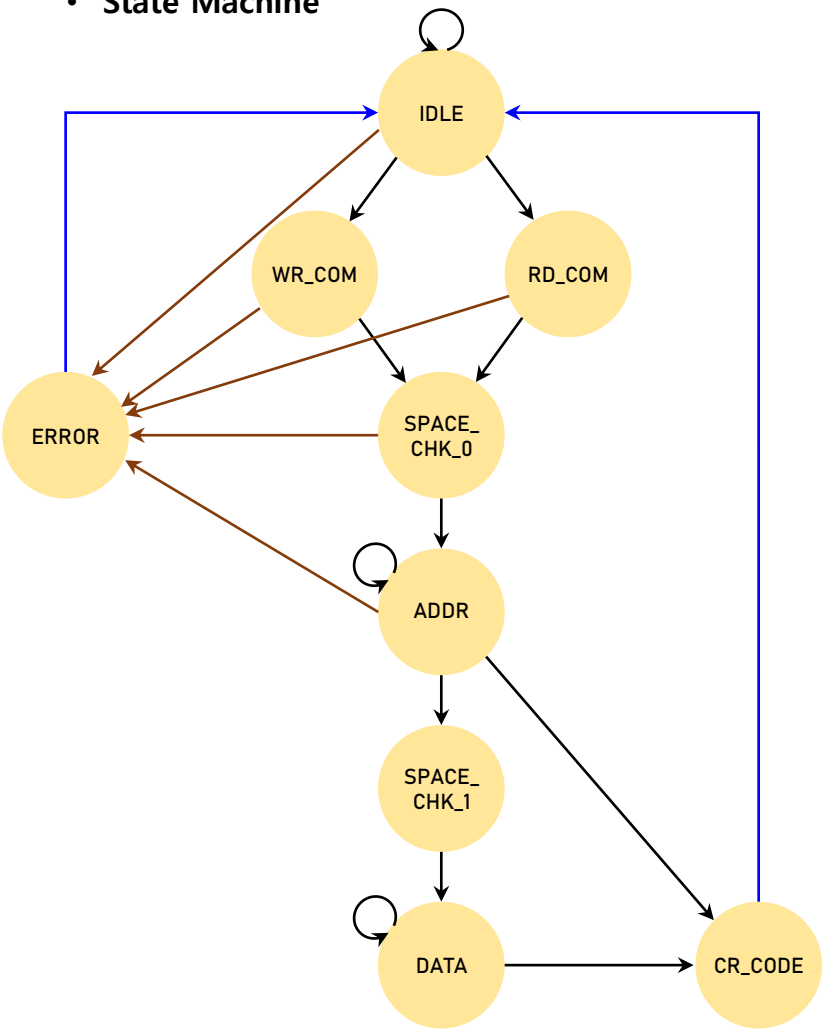
# 4. Module 설계 (uart_rx)

## com_int

- Function
  - ➢ command 분석
  - ➢ 해당 command에 따른 동작 실행
  - ➢ tx_data 생성

- Command 예제
  - ➢ command : w 1234 55aa55aa55aa
  - ➢ hex value : 0x77(w), 0x20(space), 0x31(1), 0x32(2),
    0x33(3), 0x34(4), 0x20(space), 0x35(5),
    0x35(5), 0x61(a), 0x61(a) .....
  - ➢ "w"에 따른 동작
    - ✓ Address 범위에 따른 Write 동작을 실행
      1. APB : 0x0000 – 0x0FFF
      2. UART Setting : 0xF000 – 0xFFFF
  - ➢ UART TX를 위한 데이터 생성
    - ➢ ASCII Code로 생성되어야 정상 Display됨

- State Machine



# 4. Module 설계 (uart_rx)

## ascii to hex

```
function [3:0]  ascii2nibble;
    input   [7:0]   ascii;
    reg             is_dec_a2n;
    begin
        is_dec_a2n      = (ascii[7:4] == 4'b0011) ? 1'b1 : 1'b0;
        ascii2nibble    = ascii[3:0] + (is_dec_a2n ? 4'd0 : 4'd9);
    end
endfunction
```

## hex to ascii

```
function [7:0]  nibble2ascii;
    input   [3:0]   nibble;
    reg             is_dec_n2a;
    reg     [3:0]   n2a_h;
    reg     [3:0]   n2a_l;
    begin
        is_dec_n2a      = (nibble < 4'd10) ? 1'b1 : 1'b0;
        n2a_h           = is_dec_n2a ? 4'b0011 : 4'b0100;
        n2a_l           = nibble - (is_dec_n2a ? 4'd0 : 4'd9);
        nibble2ascii    = {n2a_h,n2a_l};
    end
endfunction
```

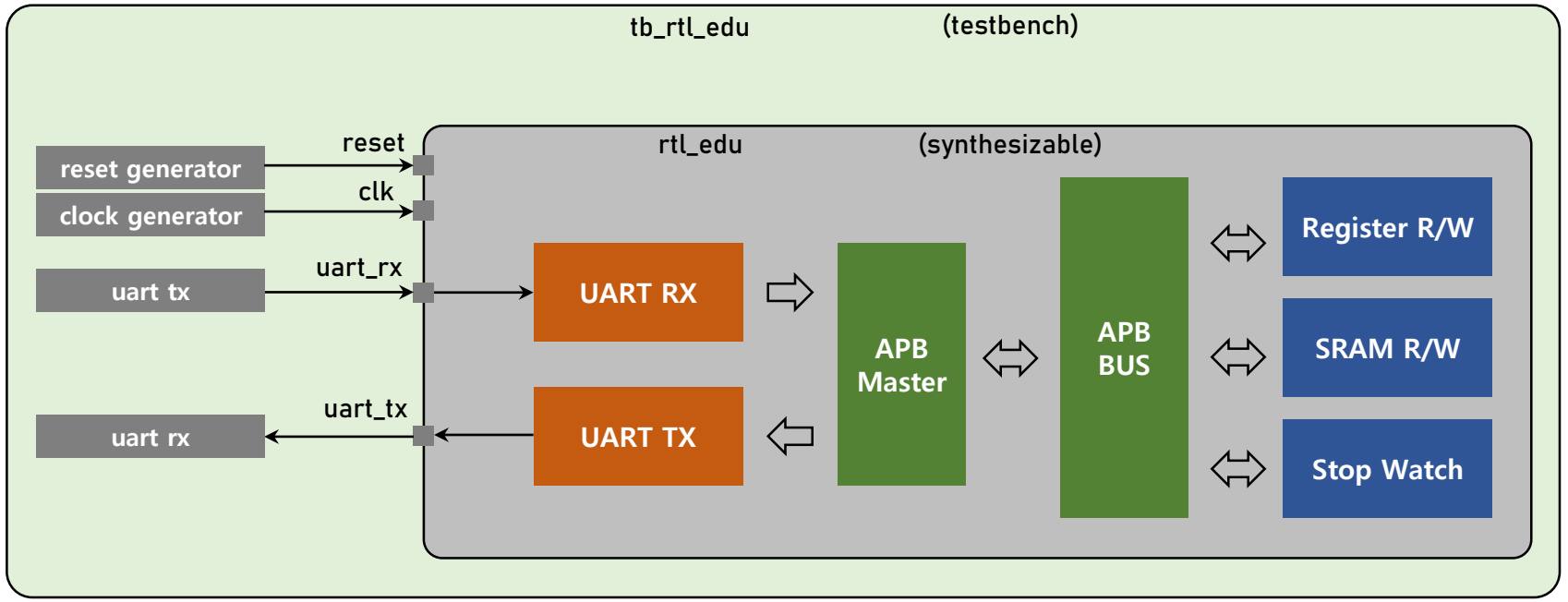
# 4. Module 설계 (uart_rx)

## Verilog Code

# 부록

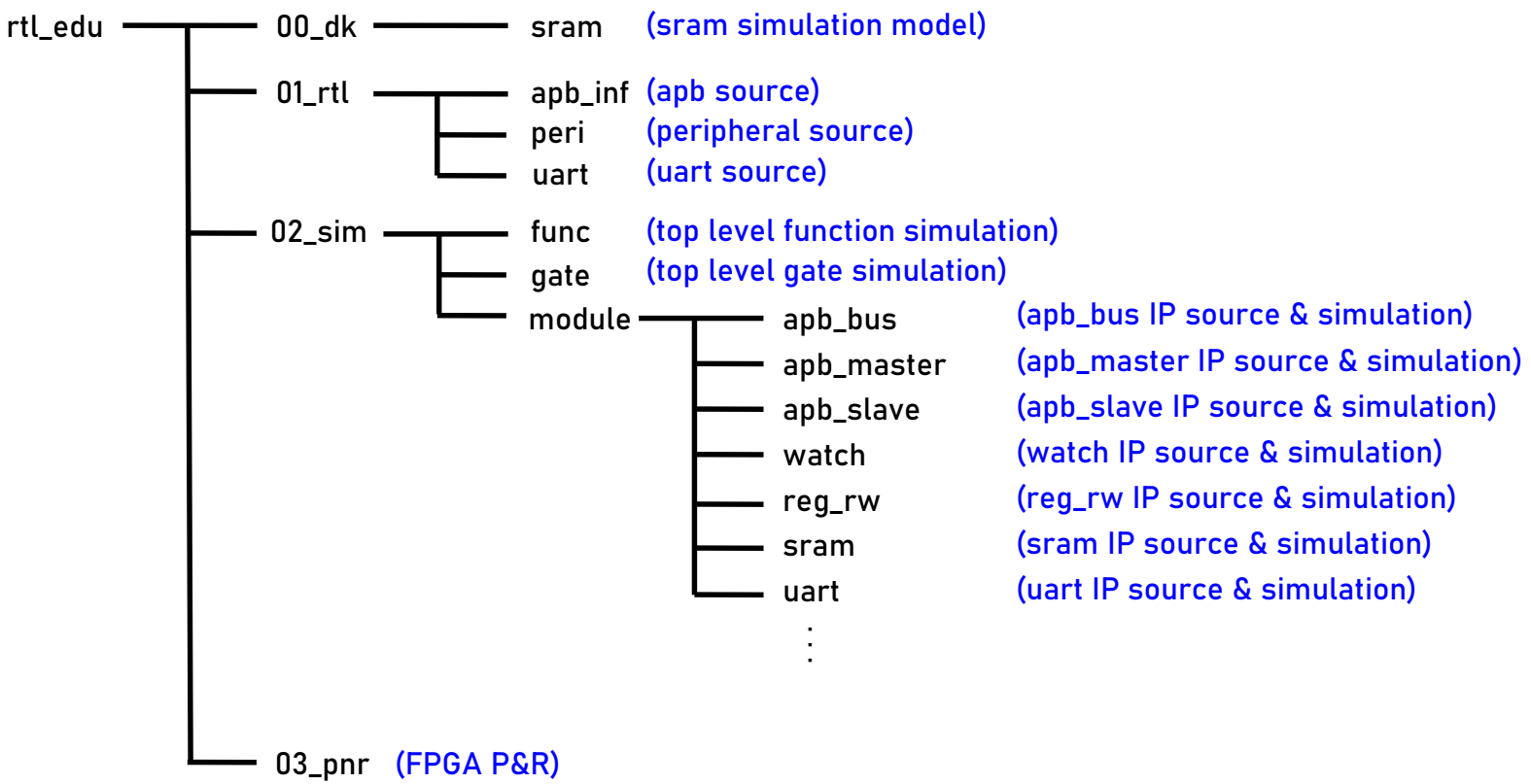
| dec | hex | oct | char | dec | hex | oct | char | dec | hex | oct | char | dec | hex | oct | char |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 000 | NULL | 32 | 20 | 040 | space | 64 | 40 | 100 | @ | 96 | 60 | 140 | ` |
| 1 | 1 | 001 | SOH | 33 | 21 | 041 | ! | 65 | 41 | 101 | A | 97 | 61 | 141 | a |
| 2 | 2 | 002 | STX | 34 | 22 | 042 | " | 66 | 42 | 102 | B | 98 | 62 | 142 | b |
| 3 | 3 | 003 | ETX | 35 | 23 | 043 | # | 67 | 43 | 103 | C | 99 | 63 | 143 | c |
| 4 | 4 | 004 | EOT | 36 | 24 | 044 | $ | 68 | 44 | 104 | D | 100 | 64 | 144 | d |
| 5 | 5 | 005 | ENQ | 37 | 25 | 045 | % | 69 | 45 | 105 | E | 101 | 65 | 145 | e |
| 6 | 6 | 006 | ACK | 38 | 26 | 046 | & | 70 | 46 | 106 | F | 102 | 66 | 146 | f |
| 7 | 7 | 007 | BEL | 39 | 27 | 047 | ' | 71 | 47 | 107 | G | 103 | 67 | 147 | g |
| 8 | 8 | 010 | BS | 40 | 28 | 050 | ( | 72 | 48 | 110 | H | 104 | 68 | 150 | h |
| 9 | 9 | 011 | TAB | 41 | 29 | 051 | ) | 73 | 49 | 111 | I | 105 | 69 | 151 | i |
| 10 | a | 012 | LF | 42 | 2a | 052 | * | 74 | 4a | 112 | J | 106 | 6a | 152 | j |
| 11 | b | 013 | VT | 43 | 2b | 053 | + | 75 | 4b | 113 | K | 107 | 6b | 153 | k |
| 12 | c | 014 | FF | 44 | 2c | 054 | , | 76 | 4c | 114 | L | 108 | 6c | 154 | l |
| 13 | d | 015 | CR | 45 | 2d | 055 | - | 77 | 4d | 115 | M | 109 | 6d | 155 | m |
| 14 | e | 016 | SO | 46 | 2e | 056 | . | 78 | 4e | 116 | N | 110 | 6e | 156 | n |
| 15 | f | 017 | SI | 47 | 2f | 057 | / | 79 | 4f | 117 | O | 111 | 6f | 157 | o |
| 16 | 10 | 020 | DLE | 48 | 30 | 060 | 0 | 80 | 50 | 120 | P | 112 | 70 | 160 | p |
| 17 | 11 | 021 | DC1 | 49 | 31 | 061 | 1 | 81 | 51 | 121 | Q | 113 | 71 | 161 | q |
| 18 | 12 | 022 | DC2 | 50 | 32 | 062 | 2 | 82 | 52 | 122 | R | 114 | 72 | 162 | r |
| 19 | 13 | 023 | DC3 | 51 | 33 | 063 | 3 | 83 | 53 | 123 | S | 115 | 73 | 163 | s |
| 20 | 14 | 024 | DC4 | 52 | 34 | 064 | 4 | 84 | 54 | 124 | T | 116 | 74 | 164 | t |
| 21 | 15 | 025 | NAK | 53 | 35 | 065 | 5 | 85 | 55 | 125 | U | 117 | 75 | 165 | u |
| 22 | 16 | 026 | SYN | 54 | 36 | 066 | 6 | 86 | 56 | 126 | V | 118 | 76 | 166 | v |
| 23 | 17 | 027 | ETB | 55 | 37 | 067 | 7 | 87 | 57 | 127 | W | 119 | 77 | 167 | w |
| 24 | 18 | 030 | CAN | 56 | 38 | 070 | 8 | 88 | 58 | 130 | X | 120 | 78 | 170 | x |
| 25 | 19 | 031 | EM | 57 | 39 | 071 | 9 | 89 | 59 | 131 | Y | 121 | 79 | 171 | y |
| 26 | 1a | 032 | SUB | 58 | 3a | 072 | : | 90 | 5a | 132 | Z | 122 | 7a | 172 | z |
| 27 | 1b | 033 | ESC | 59 | 3b | 073 | ; | 91 | 5b | 133 | [ | 123 | 7b | 173 | { |
| 28 | 1c | 034 | FS | 60 | 3c | 074 | < | 92 | 5c | 134 | \ | 124 | 7c | 174 | \| |
| 29 | 1d | 035 | GS | 61 | 3d | 075 | = | 93 | 5d | 135 | ] | 125 | 7d | 175 | } |
| 30 | 1e | 036 | RS | 62 | 3e | 076 | > | 94 | 5e | 136 | ^ | 126 | 7e | 176 | ~ |
| 31 | 1f | 037 | US | 63 | 3f | 077 | ? | 95 | 5f | 137 | _ | 127 | 7f | 177 | DEL |

## System Block Diagram



# 부록

## Directory Structure



# Thank you

CoAsia SEMI Ltd.