

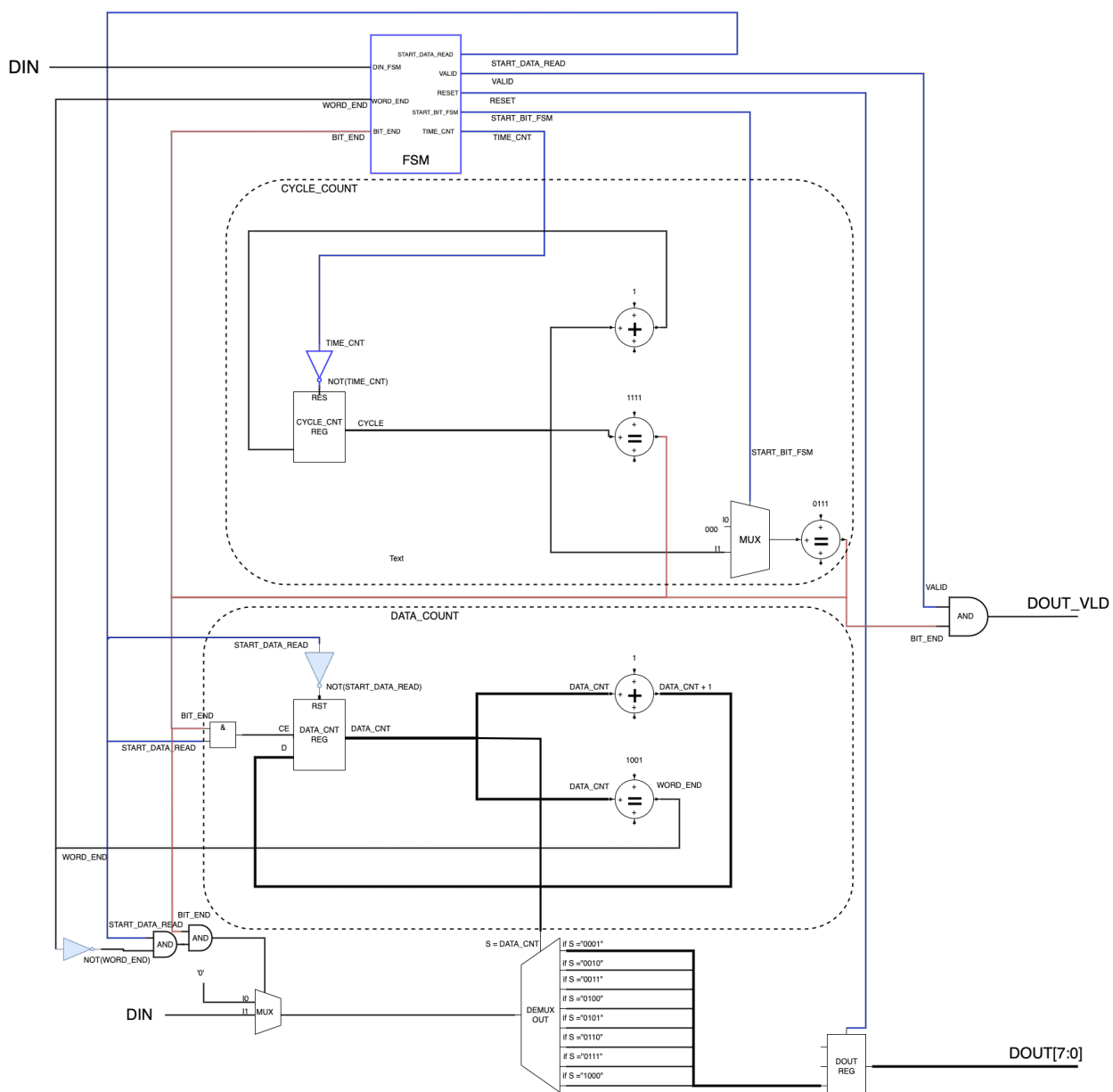
Výstupní zpráva

Jméno: Pavel Stepanov

Login: xstepa77

Architektura navrženého obvodu (na úrovni RTL)

Schéma obvodu



Obr. č. 1. - schema obvodu UART_RX na úrovni RTL

Popis funkce

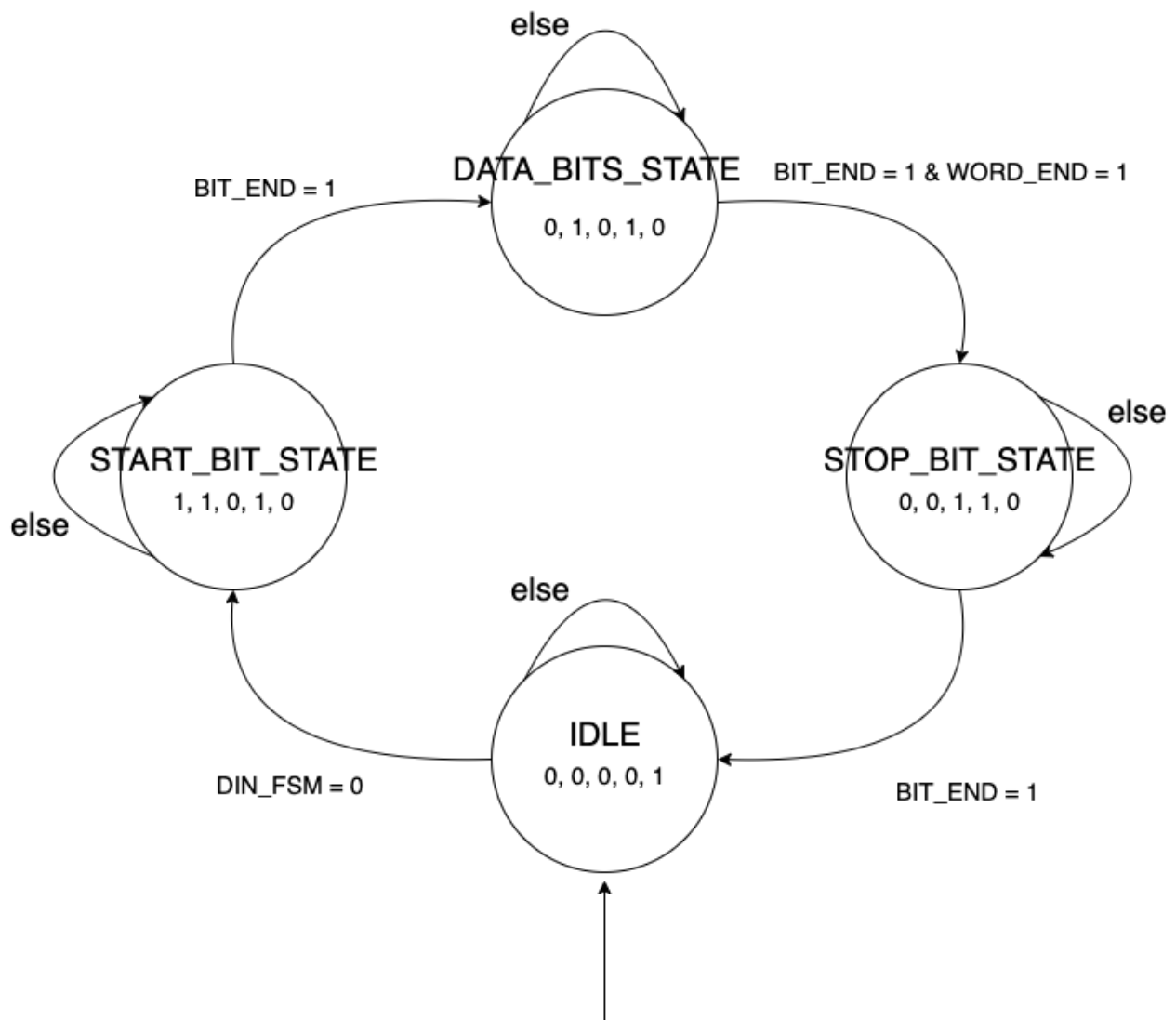
Během IDLE stavu zařízení nedělá nic. Pokud však na vstupu DIN přijde signál 0, spustí se čítač CLK pomocí FSM (TIME_CNT). Nejprve se spočítá 8 CLK cyklů, což umožní přečíst middle bit na start bitu. V opačném případě bude během procesu TIME_COUNT spočteno 16 CLK cyklů. Když se spočítá požadovaný počet CLK cyklů, nastaví se BIT_END = 1 (1 cyklus pro čtení). Poté se pomocí FSM (START_BIT) a BIT_END spustí počítání čtených datových bitů (proces DATA_COUNT). Přečte se 10 datových bitů - start bit (DATA_CNT = "0000"), 8 datových bitů ("0001" - "1000") a stop bit ("1001"). Data "0001" - "1000" se rozdělí pomocí 8:1 demultiplexoru do osmibitového registru, který bude výstupem zařízení (DOUT[7:0]). Poté se signál WORD_END oznamující, že data byla přečtena, nastaví na hodnotu 1. Použitím FSM (VALID) a čítače hodin (BIT_END) nastavíme signál DOUT_VLD na 1 (pro DOUT_VLD je potřeba, aby BIT_END bylo 16 CLK cyklů od prostředního bitu stop bitu a aby signál trval přesně 1 CLK cyklus). Poté zařízení přejde do IDLE stavu pomocí FSM a na výstupů kterého budeme mít RESET = 1. Ten signal nastává osmibitový registr DOUT[7:0] na "0000000".

Návrh automatu (Finite State Machine)

Schéma automatu

Legenda:

- Stavy automatu: IDLE, START_BIT_STATE, DATA_BITS_STATE, STOP_BIT_STATE
- Vstupní signály: DIN_FSM, BIT_END, WORD_END
- Mealyho výstupy: žádný
- Moorovy výstupy: START_BIT_FSM, START_DATA_READ, VALID, TIME_CNT, RESET



Obr. č. 2. - schema automatu

Popis funkce

Počáteční stav automatu = IDLE. Ve stavu IDLE automat čeká na signal $DIN_FSM = DIN = 1$, všechny výstupy jsou 0 (kromě $RESET = 1$), který ukazuje, že dostali jsme start bit. Přechází ve stav $START_BIT_STATE$, který na výstup nastaví na 1 signály $TIME_CNT$ (pro počítání 16 CLK cyklů) $START_BIT_FSM$ (který potřebujeme pro spočítání 8 CLK cyklů, aby zařízení četl middle bit) a $START_DATA_READ$, pomocí kterého se spustí process $DATA_COUNT$. Potom po 1 BIT_END cyklu přejde do stavu $DATA_BITS_STATE$ ve kterém mění na 0 $START_BIT_FSM$. Když přečte 10 bit ($WORD_END = 1$) a budeme na middle stop bitu, automat přejde do stavu $STOP_BIT_STATE$ a nastaví $START_DATA_READ$ na 0 a $VALID$ na 1. Pak po 1 BIT_END cyklu přejde ve stav IDLE. V ostatních případech se stavy nebudou měnit.

Snímek obrazovky ze simulací

