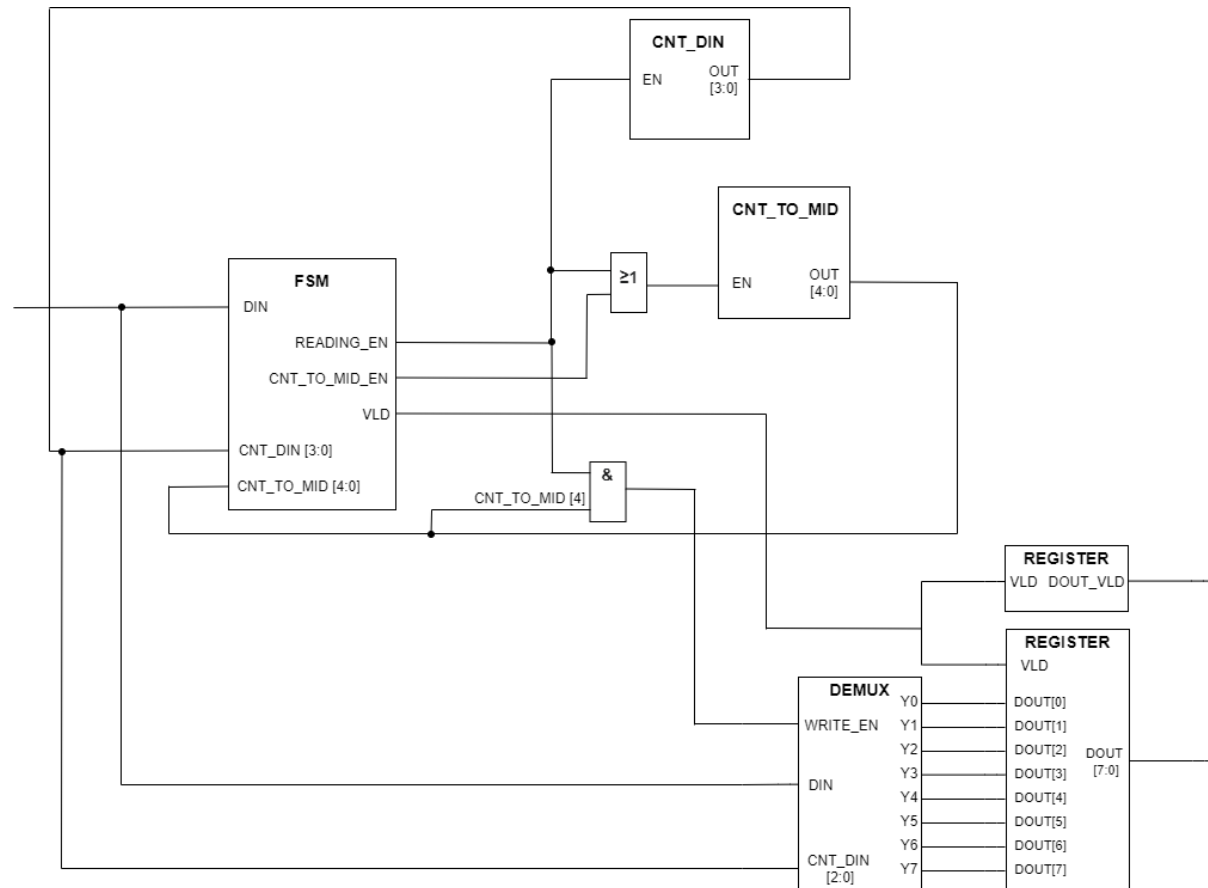


Meno: Patrik Sehnoutek

Login: xsehno01

Architektúra navrhnutého obvodu (na úrovni RTL)

Schéma obvodu



Popis funkcie

Môj vyššie navrhnutý RTL obvod sa skladá z konečného automatu (FSM), dvoch počítadiel (CNT_DIN, CNT_TO_MID), AND hradla, OR hradla, demultiplexoru a dvoch registrov.

Konečný automat (FSM) – Obsahuje 3 vstupy (signál DIN, vektory CNT_DIN, CNT_TO_MID) a 3 výstupy (signály READING_EN, CNT_TO_MID_EN a VLD). Má 4 stavy, ktoré sa menia na základe jeho vstupov, následne sú jeho výstupy odvodené od súčasného stavu. Na základe výstupov sa riadi zvyšok obvodu.

Počítadlo (CNT_TO_MID) – počítadlo, ktoré nám slúži na počítanie stredov daných bitov, tzv. „midbitov“.

Počítadlo (CNT_DIN) – počítadlo, ktoré nám slúži na počítanie načítaných bitov.

Demultiplexor (DEMUX) – obsahuje 3 vstupy (signály WRITE_EN a DIN, vektor CNT_DIN) a 8 výstupov (jednotlivé bity výstupného vektoru DOUT, ktoré sa uložia do registra). WRITE_EN je nastavený na hodnotu '1' práve vtedy, keď je konečný automat v stave čítania dát a nachádzame sa v strede čítaného bitu, k čomu nám slúži počítadlo

CNT_TO_MID. Zápis vstupu na správnu pozíciu na výstupe zariadení adresové vstupy, na ktoré je pripojený vektor CNT_DIN, ktorý počíta načítané vstupné bity.

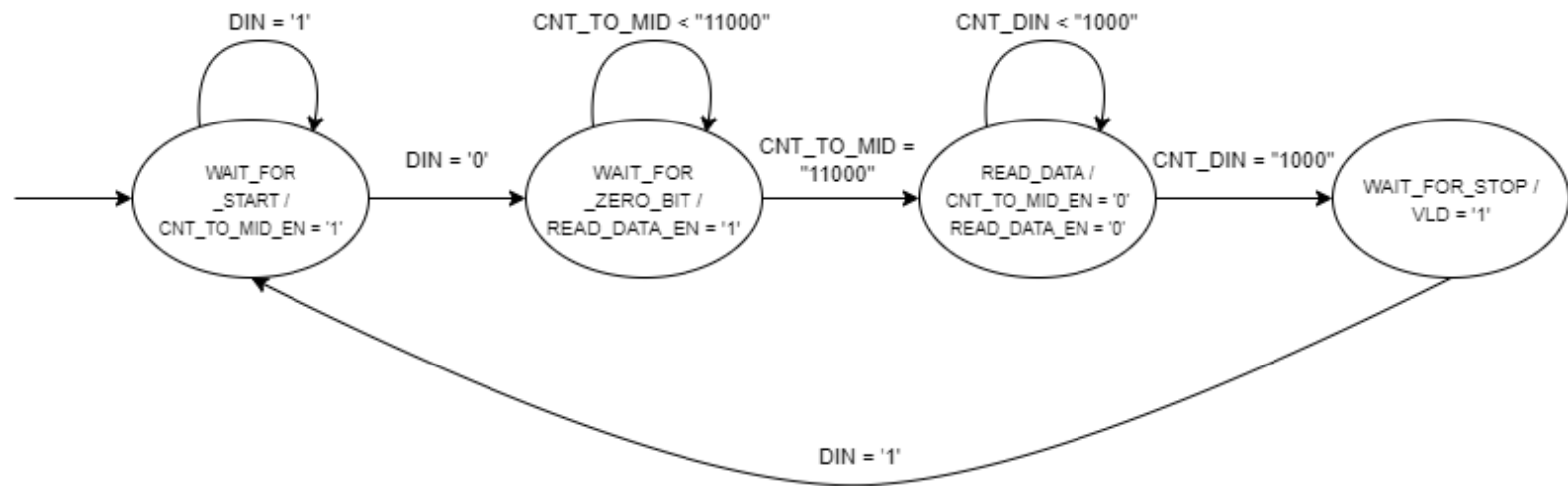
Registre – register vyššie uchováva hodnotu signálu DOUT_VLD, register nižšie uchováva hodnotu výstupného vektoru DOUT. Výstup z registrov je riadený signálom VLD z konečného automatu. Ak bude signál VLD nastavený na '1' na výstupe budú validné dáta, ktoré by sa dali následne ďalej spracovávať.

Návrh automatu (Finite State Machine)

Schéma automatu

Legenda:

- **Stavy automatu:** WAIT_FOR_START, WAIT_FOR_ZERO_BIT, READ_DATA, WAIT_FOR_STOP
- **Vstupné signály:** DIN, CNT_DIN, CNT_TO_MID
- **Moorove výstupy:** CNT_TO_MID_EN, READ_DATA_EN, VLD



Popis funkcie

Môj vyššie navrhnutý konečný automat obsahuje 4 stavy:

1. **WAIT_FOR_START** – čakanie na START bit
2. **WAIT_FOR_ZERO_BIT** – čakanie na nultý bit
3. **READ_DATA** – načítavanie vstupu a ukladanie na výstup
4. **WAIT_FOR_STOP** – čakanie na STOP bit

WAIT_FOR_START – počiatočný stav konečného automatu. Na prechod do nasledujúceho stavu je potrebné, aby na vstupe (DIN) bola '0', čo nám signalizuje START bit. Pokiaľ bude na vstupe (DIN) '1', tak sa stav nezmení. Pri zmene stavu sa nastaví signál CNT_TO_MID_EN na '1', čím sa spustí počítadlo hodinového signálu.

WAIT_FOR_ZERO_BIT – druhý stav konečného automatu. Na prechod do nasledujúceho stavu je potrebné, aby počítadlo CNT_TO_MID malo hodnotu "11000" (24). Počítadlo CNT_TO_MID musí byť rovné 24, lebo do každého vstupného bitu sa zmestí 16 hodinových cyklov a jednotlivé bity máme čítať v „midbite“ podľa zadania. Pri zmene stavu sa nastaví signál READ_DATA_EN na '1', čím sa spustí počítadlo načítaných bitov.

READ_DATA – tretí stav konečného automatu. Na prechod do nasledujúceho stavu je potrebné, aby počítadlo CNT_DIN malo hodnotu "1000" (8). Podľa zadania obsahuje jedno slovo 8 bitov, preto načítavame 8 bitov. Signály CNT_TO_MID_EN a READ_DATA_EN sa nastaví do '0'.

WAIT_FOR_STOP – posledný stav konečného automatu. Na prechod do prvého stavu je potrebné, aby na vstupe (DIN) bola '1', čo nám signalizuje STOP bit. Pri zmene stavu sa nastaví signál VLD na hodnotu '1', čo nám signalizuje správnosť výstupu.

Help

100 ps

Layout Simulate

ColumnLayout Prefault

Instance	Design unit	Design unit type	Visibility
testbench	testbench(tb)	Architecture	+acc=...
dut	uart_rx(tbe...)	Architecture	+acc=...
clk_process	testbench(tb) Process		+acc=...
test	testbench(tb) Process		+acc=...
standard	std_logic_1164 Package		+acc=...
std_logic_1164	std_logic_1164 Package		+acc=...
std_logic_arith	std_logic_arith Package		+acc=...
std_logic_unsigned	std_logic_unsigned Package		+acc=...

Name	Value	Kind
clk	0	Signal
rst	0	Signal
dn	1	Signal
dout	01010101	Signal
dout_wld	0	Signal

Global Signals	Mags
rst	1
clk	0
dn	1
dout	00000000
dout_wld	0

Name	Type (filtered)	State
Processes (Active)		
Or		

