

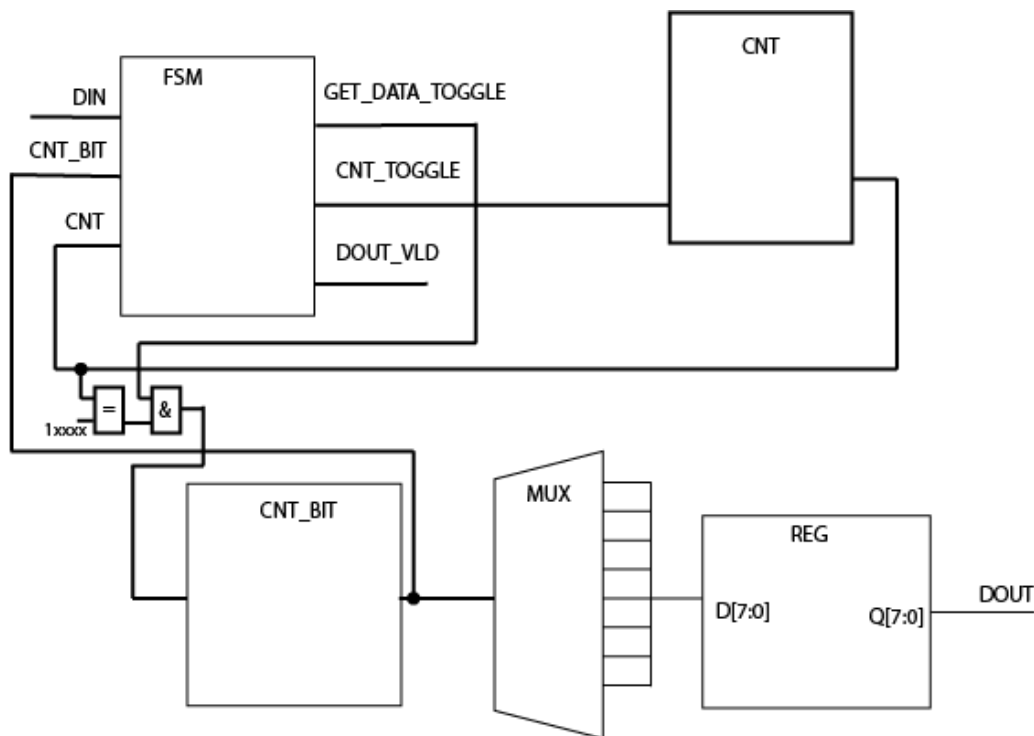


Výstupní zpráva INC projekt

Tomáš Souček
xsouce15

Architektura navrženého obvodu (na úrovni RTL)

Schéma obvodu

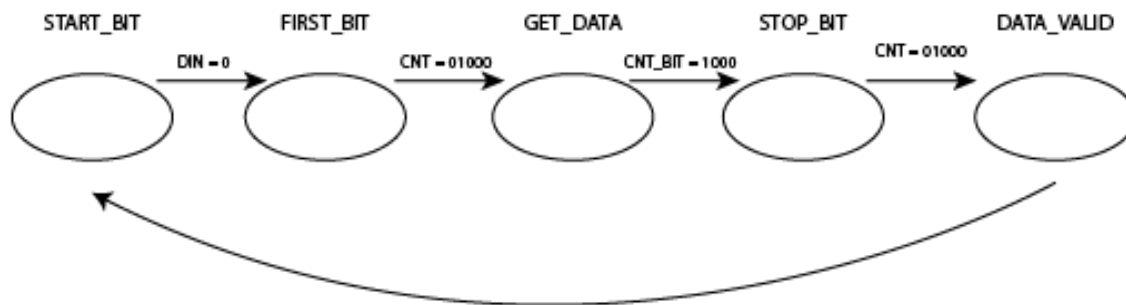


Popis funkce

FSM na vstup dostane CLK, RST, DIN (vstupní data), CNT_BIT (počet přečtených bitů) a CNT (počet hodinových signálů). CLK a RST jsem na základě zadání nezakresloval. Detailnější popis FSM je níže. Pokud se obvod nachází ve fázi, kdy chci mít přehled o aktuálním počtu hodinových signálů, CNT počítá hodinové signály a když je CNT číslo větší než 16 (obsahuje na MSB 1), pak projde prvním komparátorem na cestě ke čtení dat. Pokud se obvod současně nachází ve stavu čtení dat (GET_DATA_TOGGLE), spustí se počítání přečtených bitů. Počet přečtených bitů se zasílá do FSM, kde se čeká na přečtení všech 8 bitů, aby se obvod mohl posunout do další fáze. Při čtení bitů zároveň rovnou dochází k zápisu dat přes multiplexor (MUX) a registry do DOUT.

Návrh automatu (Finite State Machine)

Schéma automatu



Popis funkce

Na začátku architektury jsem si v kódu podle zadání navrhnul 5 stavů – počátek procesu (START_BIT), načtení prvního bitu (FIRST_BIT), čtení dat (GET_DATA), načtení posledního bitu (STOP_BIT) a potvrzení validních dat (DATA_VALID). Na začátek jsem rovnou přiřadil počáteční stav. Před začátkem procesu mám pomocné proměnné na přepínání mezi stavy.

Proces začíná kontrolou, jestli se zrovna nachází na rostoucí hraně. Pokud ano, začíná vykonávat požadované funkce. Pokud je nastaven restart na true, aktuální stav se nastaví na počáteční stav. V opačném případě začne proces pomocí switch konstrukce zjišťovat ve kterém stavu se momentálně nachází a jakou podmínku musí splnit, pro postup do dalšího stavu.

Snímek obrazovky ze simulací

