

PRÁCA S MEGACORE FUNKCIAMI - FIR A IIR OD SPOLOČNOSTI ALTERA

Na cvičení si ukážeme postup pri návrhu FIR a IIR filtrov pre implementáciu v hradlových poliach pomocou MegaCore funkcií od spoločnosti Altera.

1. IIR kompilátor

IIR kompilátor od spoločnosti Altera možno použiť na implementáciu kaskádnych a paralelných štruktúr a IIR rádu 1 až 15. Podporuje priamu podporu pre implementáciu v najnovších obvodoch Stratix a Cyclone, automatickú konverziu na kaskádnú a paralelnú štruktúru, šírka dát a koeficientov môže byť 1 až 32 bitov.

Pre získanie prehľadu o počte použitých LE a využití DSP blokov (boli predstavené na prednáške), uvediem príklad implementácie IIR filtra 4. rádu (Čebyševov pásmový priepust, 18-bitové koeficienty, 12-bitové dáta, II. priama forma)

- pri použití obvodov APEX II je potrebných 1129 LE,
- pri obvodoch Stratix sú využité 3 DSP bloky a len 124 LE!

V ďalších riadkoch nasleduje opis ako vytvoriť IIR filter s požadovanými parametrami.

Otvoríme program Quartus II a vytvoríme nový projekt - je potrebné špecifikovať pracovný adresár (cvikoX) a názov projektu. V ďalšom okne klikneme na *User Library Pathnames* a následne na *Add* a nastavíme cestu: *C:\altera\MegaCore\iir-v1.3.2\lib*. Vytváranie ukončíme klikom na *Finish*.

Po vytvorení projektu môžeme spustiť samotnú MegaCore funkciu. V menu *Tools* zvolíme *MegaWizard Plug-In Manager* a vyberieme prvú položku, teda vytvorenie novej megafunkcie. Otvorí sa okno s ponukou LPM funkcií a takisto s nami požadovaným IIR kompilátorom. V ľavej časti okna pod *Signal Processing, Filters*, klikneme na *IIR Filter*. Zvolíme VHDL (v pravej časti okna) a zvolíme názov filtra. Následne sa spustí IIR kompilátor a my môžeme voliť parametre implementovaného filtra.

Môžeme si vybrať priamu formu, kaskádnú formu, alebo paralelnú formu a rád filtra. K tomuto oknu sa možno kedykoľvek vrátiť. Ak sme medzitým už zadali koeficienty filtra pre určitú formu, po jej zmene sa koeficienty automaticky prepočítajú.

V ďalšom okne môžeme nastaviť koeficienty filtra a to dvojakým spôsobom. Buď zadávame koeficienty priamej a spätnej vetvy (*a* a *b*) alebo možno filter definovať cez definovanie polohy pólov a núl.

Keďže hodnoty koeficientov sú často vyjadrené cez čísla s plávajúcou rádovou čiarkou, treba ich zmeniť na čísla s pevnou rádovou čiarkou, a to vynásobením škálovacím faktorom a zaokrúhlením. To sa môže realizovať viacerými spôsobmi:

- škálovanie na určitý počet bitov,
- ohraničenie škálovacích faktorov na mocniny 2,
- manuálne škálovanie,
- žiadne škálovanie (koeficienty už sú škálované).

Tiež možno zvoliť oneskorenie výstupu voči vstupnej vzorke, keď máme 3 možnosti: nulové oneskorenie - plne kombinačná logika, stredné oneskorenie - registrovaná verzia - oneskorenie 1 hodnový takt, maximálne oneskorenie - závisí od rádu filtra - plne zreťazená verzia. Poslednou voľbou v tomto okne je spôsob implementácie. Ak zaškrtneme *Use dedicated Circuitry*, pri použití obvodov Stratix budú využité DSP bloky. Druhou voľbou je možnosť použitia distribuovanej aritmetiky.

V šiestom okne možno upraviť niektoré parametre pre implementáciu a overiť si odpovede filtra v časovej a frekvenčnej oblasti na rôzne vstupné signály - impulz, biely šum...

V predposlednom okne možno zvoliť nastavenia pre simuláciu. Pre kompilátor IIR je možné vygenerovať len model pre Matlab a zvoliť si typ stimulu - vstupného signálu pre simuláciu v programe Quartus.

V poslednom okne už len potvrdíme vytvorenie súborov. Spôsob kompilácie a simulácie filtra si ukážeme pri FIR filtri.

2. FIR kompilátor

Poskytuje priamu podporu pre nové obvody Stratix a Cyclone, umožňuje vytvárať širokú škálu architektúr filtrov

- s pevne zadanými koeficientami - plne paralelné, sériové, viac-bitové sériové, podpora interpolácie a decimácie
- variabilné filtre - viaccyklové, podpora viacerých sád koeficientov.

Nasleduje podobný postup ako pri IIR filtroch, teda otvoríme program Quartus II a vytvoríme nový projekt - je potrebné špecifikovať pracovný adresár (cviKoX) a názov projektu. V ďalšom okne klikneme na *User Library Pathnames* a následne na *Add* a nastavíme cestu: *C:\altera\MegaCore\fir_compiler-v2.6.3\lib_time_limited*. Vytváranie ukončíme klikom na *Finish*.

Po vytvorení projektu môžeme spustiť samotnú MegaCore funkciu. V menu *Tools* zvolíme *MegaWizard Plug-In Manager* a vyberieme prvú položku, teda vytvorenie novej megafunkcie. Otvorí sa okno s ponukou LPM funkcií a takisto s nami požadovaným FIR kompilátorom. V ľavej časti okna pod *Signal Processing, Filters*, klikneme na *FIR Filter*. Zvolíme VHDL (v pravej časti okna) a zvolíme názov filtra totožný s názvom projektu (toto musí byť bezpodmienečne dodržané!!!). Následne sa spustí FIR kompilátor a my môžeme voliť parametre implementovaného filtra.

V prvom okne môžeme zvoliť koeficienty filtra, na čo slúžia 3 spôsoby: koeficienty môžeme generovať pomocou FIR generátora, koeficienty môžu byť vyčítané zo súboru - filter možno navrhnuť v iných programoch a následne importované do FIR kompilátora, alebo môžeme vytvoriť sada nulových koeficientov pri použití variabilného filtra, keď sa koeficienty generujú v inej aplikácii.

Ako príklad navrhnete filter s týmito parametrami:

typ filtra: band pass (pásmový pripust), typ okna: Hamming, vzorkovacia frekvencia: 50e+006, počet koeficientov: 77, hraničné frekvencie: 5e+006, 10e+006.

Po zadání parametrov a kliknutí na *Apply* sa zobrazí frekvenčná odpoveď filtra červenou farbou a hodnoty koeficientov modrou farbou.

Keďže hodnoty koeficientov sú často vyjadrené cez čísla s plávajúcou rádovou čiarkou, treba ich zmeniť na čísla s pevnou rádovou čiarkou. Škálovanie koeficientov sa nastavuje v ďalšom okne. Zároveň možno na grafe sledovať vplyv škálovania koeficientov na prenosové charakteristiky filtra.

V ďalšom okne špecifikujeme typ a šírku vstupných dát a takisto výstupných dát.

Okno *Architecture* nám dáva možnosť nastaviť spôsob implementácie FIR filtra. Môžeme voliť typ obvodov pre implementáciu, štruktúru filtra. Pri voľbe typu filtra si všimnime základné vlastnosti:

- paralelná architektúra - veľká plocha, rýchla filtrácia
- sériová štruktúra - malá plocha, vyžaduje viac taktov pre jeden výpočet
- zret'azenie - zväčšením plochy sa získa vysoký výkon filtra, priepustnosť sa zvýši za cenu pridania oneskorenia

Pri zmene typu obvodov a štruktúry filtra môžeme sledovať v časti *Resource* aké má daný filter nároky na počet logických elementov, pamäťové bloky a DSP bloky. Navyše možno meniť aj spôsob uloženia koeficientov a dát - v pamäťových blokoch alebo v logike.

V nasledujúcom okne možno celý navrhnutý filter "odsimulovať". Možno meniť tvary vstupných signálov a sledovať výstupné priebehy. Zároveň možno zvoliť, ktoré modely chceme vygenerovať pre simuláciu.

V adresári s projektom sa objaví súbor *názov_filtra_rpt.htm*. Ten obsahuje súhrn informácií o vytvorenom filtre a súboroch pre jeho simuláciu.

V programe Quartus klikneme na *Start compilation* (bordová šípka v hornom rade ikon). Po úspešnej kompilácii môžeme skontrolovať výsledky implementácie, čiže počet obsadených logických a pamäťových elementov a maximálnu frekvenciu hodinového signálu.

Pre spustenie simulácie musíme nastaviť parametre simulácie - v ponuke *Assignments* v *Settings* nájdeme položku *Simulator Settings*. V časti *Mode* nastavíme simulačný mód na *Functional* a v časti *Time/Vectors* nastavíme *Source of vector stimuli* na súbor *názov_projektu.vec*. Potvrdíme zmenu *OK*. A spustíme simuláciu *Run Simulation* (modrá šípka). Po skončení simulácie je možné si prezrieť získané výstupy. Žiaľ spoločnosť Altera neposkytuje vytvorenie testbenchu pre program ModelSim, ktorý sme používali na predchádzajúcom cvičení, preto sme pre simuláciu použili program Quartus.

Pre získanie porovnania medzi rôznymi implementáciami filtrov môžeme celý postu opakovať a meniť typ architektúry filtra, využiť DSP bloky Stratix obvodov, zvoliť rôzne typy vstupných signálov pre simuláciu a podobne.