Použité transformácie

Pre frekvenciu čipu 100 Mhz hľadáme riešenie, kde II (inicializačný interval) musí byť 1 (100 / 100) aby bola splnená podmienka, že vstupné vzorky sú spracovávané požadovanou vzorkovacou frekvenciou 100 Mhz, a počet tapov bude najvyšší. II nastavený na 1 znamená, že medzi každou iteráciou cyklu je oneskorenie jedného hodinového cyklu. Základné jednoduché riešenie so 4 tapmi má latenciu 14, II je 14, využitie čipu je 1%. Po zvýšení počtu tapov na 32 sa zvýšila latencia na 130 cyklov, II je 130, využitie čipu ostáva na 1%. Po úplnom rozbalení oboch cyklov je latencia 17, II je 17. V tabuľke *Utilization Estimates* je uvedené, že počet dostupných DSP je 66. Momentálne sa používa 32, čo odpovedá aktuálne nastavenému počtu tapov. Následný pokus so zmenou počtu tapov potvrdzuje predpoklad, že 67 tapov už je príliš veľa kvôli nedostatočným zdrojom čipu a maximum je 66 tapov. Ďalej som aplikoval techniku *array partitioning* na polia h a regs. Najlepšie výsledky som dosiahol v režime *complete*, kde prvky poľa sú rozdelené do nezávislých registrov. Latencia je 33, II je 33. Doterajšie experimenty sa k II = 1 nepriblížili, a preto po zistení tejto hraničnej hodnoty počtu tapov som aplikoval ďalšiu techniku - zreťazenie cyklov (*loop pipelining*). V Ganttovom diagrame som zistil problém s *tree balacing* a MAC cyklus som prepísal do dvoch cyklov MUL a MAC. Po tejto úprave kódu je už *tree balacing* správny. Zreťazenie cyklov individuálne, na každom cykle zvlášť, neprinášalo lepšie výsledky, no na úrovni funkcie áno. Latencia v tomto prípade bola 65, II je 1. Znovu som aplikoval techniku *array partitioning* na polia h, regs a mult_temp v režime *complete*. Latencia pri zreťazení je 22, II sa oproti rozbaleným cyklom znížil z 33 na 1. Odhadovaná perióda hodín je 8,132 ns. Pre frekvencii čipu 100 Mhz je najvyšší možný počet tapov 66 a táto hranica je stanovená počtom dostupných DSP na čipe.

Varianta	Latencia	Interval
základne riešenie	332	332
rozbalené cykly	66	66
rozbalené cykly + *array partitioning (complete)*	33	33
zreťazené cykly	65	66
zreťazené cykly + *array partitioning (complete)*	22	1

Tabuľka 1: Vplyv optimal. techník na výslednú latenciu a interval pri 66 tapoch a frekvencii čipu 100 Mhz

zreťazené cykly + *array partitioning (complete)*	BRAM_18K	DSP48E	FF	LUT	URAM
Použité	0	66	7757	4725	0
Dostupné	100	66	28800	14400	0
Využitie (%)	0	100	26	32	0

Tabuľka 2: Odhad využitia zdrojov u nájdeného najlepšieho riešenia pri 66 tapoch a frekvencii čipu 100 Mhz

Následne som zmenil frekvenciu čipu na 300 Mhz. Aby sa vstupné vzorky spracovávali požadovanou vzorkovaciu frekvenciou, je nutné dosiahnuť II = 3 (300 / 100). Techniku zreťazenia cyklov je teda nutné aplikovať s explicitne nastaveným II na 3, a to napr. pomocou pragmy *pipeline II=3*. Toto riešenie avšak ešte nie je optimálne - nedochádza k zdieľaniu zdrojov. Možným riešením, ako prinútiť Vivado HLS aby zdieľalo zdroje, je použiť pragmu *allocation* a pomocou nej obmedziť počet inštancií MUL na 66 (počet všetkých DSP na čipe). Predpoklad na základe teórie hovorí, že pri II = 3 spolu so zdieľanými zdrojmi je možné dosiahnuť maximálne až 198 (66 * 3) tapov. Tento predpoklad syntéza potvrdila - 198 je maximálny počet tapov pri II = 3, zistená latencia je 68. Pri 199 tapoch už Vivado HLS nedokázalo splniť podmienku, aby II bol 3, a muselo upraviť II na 4. Pri II = 4 a frekvencii čipu 300 Mhz by sme už avšak nespracovávali vzorky požadovanou vzorkovacou frekvenciou. Pri frekvencii čipu 300 Mhz je maximálny možný počet tapov 198.

Vhodným nastavením rozhrania u vstupného poľa h je ap_memory (toto nastavenie už napríklad Vivado HLS samo automaticky robí). Po aplikovaní techniky *array partitioning* sa zároveň upravili aj rozhrania. Po získaní finálneho riešenia je možné nastaviť *block level protocol* na ap_ctrl_none. Pre overenie funkčnosti riešenia som si vygeneroval IP core s povolenou možnosťou Synthesis and Place & Route. Výsledky ukázali, že časovanie je správne.

Porovnanie SW a HW implementácie

Na ARM procesore je možné implementovať najviac 7 tapov (766/100 = 7,66) s tým, že pre jednoduchosť je réžia SHIFT cyklu zanedbaná. V FPGA logike sa podarilo naimplementovať 66 tapov pri frekvencii čipu 100Mhz, 198 tapov pri 300Mhz. Ďalšie navýšenie je blokované počtom DSP. Maximálny počet tapov FIR filtra dosiahnutých pomocou Xilinx FPGA ZynQ je niekoľkonásobne vyšší ako v riešení pomocou ARM procesora.