

Xilinx Vivado HLS gyakorlat

Megj.: Vivado HLS alapú megoldás, a Vitis HLS alap direktívái (és így részben a feladatok is) ettől eltérőek!

C implementáció és testbench

1. Töltse le a tárgy honlapjáról a gyakorlathoz tartozó file-t.
2. Hozzon létre egy Vivado HLS projektet az alábbi beállításokkal.
 - a. Adja hozzá a projekthez a fir_hw.cpp és types.h fájlokat. A top function legyen fir_hw.
 - b. Testbench file-ként adja hozzá a projekthez a main.cpp, fir_sw.cpp file-okat.
 - c. Az órajel periódusideje legyen 100 ns, az FPGA pedig xc7k70tfbg484-1 (Kintex 7).
3. Definiálja a HW implementációra szánt szükséges adattípusokat az alábbiaknak megfelelően (types.h file-ban).
 - a. A minta bemenetek legyenek 24 bites kettes komplement számok 23 bitnyi tört résszel, wrap és truncate opciókkal. Típusnév: din_t.
 - b. Az együtthatók legyenek 18 bites kettes komplement számok 17 bitnyi tört résszel. A lebegőpontos értékekből kerekítéssel és wrap-olással képződjenek. Típusnév: coeff_t.
 - c. Az akkumulátor felbontása legyen a szükséges pontosságú, úgy, hogy túlcserélési hiba ne fordulhasson elő, plusz wrap és truncate. Típusnév: accu_t.
 - d. A kimenet legyen 24 bites (23 bit törtrésszel), mely az akkumulátorból csonkolással és szaturációval generálódjon. Típusnév: dout_t.
4. A fir_sw.cpp file-ba implementáljon egy FIR szűrőt float adattípus használatával. A mintatárat „tömb shifteléssel” valósítsa meg. Az együtthatókészletet a file tartalmazza (coeff_sw tömb).
5. Ellenőrizze a működést Dirac gerjesztéssel!
6. Ellenőrizze a működést egységugrás gerjesztéssel!
7. Írja meg a szűrő FPGA implementációra szánt változatát (azaz kb. copy-paste a HW implementációra szánt adattípusokkal).
8. A main() függvényben verifikálja, hogy a fixpontos adattípusokat használó megoldás megfelel az elvárásoknak. Azaz hasonlítsa össze a két implementáció hibáját ~100 mintára, s ezt írja ki a konzolra. A verifikációhoz először Dirac gerjesztést generáljon, majd pedig egy véletlen számsorozatot. A letöltött main.cpp ez utóbbit implementálja. A verifikáláshoz a Vivado HLS C debugger-t használja. Mekkora float verzióhoz képest a várt eltérés maximuma?

```

typedef ap_fixed<18, 1, AP_RND_CONV, AP_WRAP> coeff_t;
typedef ap_fixed<24, 1, AP_RND_CONV, AP_SAT> din_t;
typedef ap_fixed<18+24, 2, AP_TRN, AP_WRAP> mul_t;
typedef ap_fixed<18+24+7, 2+7, AP_TRN, AP_WRAP> accu_t;
typedef ap_fixed<24, 1, AP_TRN, AP_SAT> dout_t;

void fir_hw(din_t *smp1_din, dout_t *res)
{
    static din_t smp1[N];

    int i;
    accu_t accu;

    for_shift: for (i=127; i>=0; i--)
    {
        smp1[i] = (i==0) ? *smp1_din : smp1[i-1];
    }

    accu = 0;
    for_mac: for (i=0; i<N; i++)
    {
        accu = accu + (coeff_hw[i] * smp1[i]);
    }
    *res = accu;
}

```

1. Futtasson le egy C szintézist. Elemezze az eredményeket a felugró ablak és a Design Viewer segítségével.
 - a. A szintetizált rendszer mennyi BRAM és DSP blokkot használ? Miért?
 - b. Mekkora az egyes ciklusok és a teljes rendszer késleltetése? Miért?

Performance Estimates

Timing (ns)

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	100.00	8.454	12.50

Latency (clock cycles)

Summary

Latency		Interval		Type
min	max	min	max	
514	514	514	514	none

Detail

Instance

Loop

Loop Name	Latency		Iteration Latency	Initiation Interval		Trip Count	Pipelined
	min	max		achieved	target		
- for_shift	256	256	2	-	-	128	no
- for_mac	256	256	2	-	-	128	no

Utilization Estimates

Summary

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	1	-	-
Expression	-	-	0	223
FIFO	-	-	-	-
Instance	-	-	-	-
Memory	2	-	0	0
Multiplexer	-	-	-	81
Register	-	-	115	-
Total	2	1	115	304
Available	270	240	84400	42200
Utilization (%)	~0	~0	~0	~0

- 1x 128x18 tömb, 1x 128x24 tömb → 1-1 BRAM
- Szekvenciális végrehajtás → 1 MAC → 1 DSP, mert egy 25x18-as szorzó elegendő a 24x18 bites szorzás elvégzéséhez.
- BRAM olvasás 2 órajel → minden ciklus iteráció 2 órajel → teljes ciklus 128*2 órajel
- Ugyanez igaz a MAC ciklusra is

2. Állítsa át az órajel periódusidőt 3 ns-ra (solution1, jobb klikk, Solution Settings/Synthesis), s futtassa újra a szintézist. Mit tapasztal (erőforrásigény, késleltetés)?

Performance Estimates

▢ **Timing (ns)**

▢ **Summary**

Clock	Target	Estimated	Uncertainty
ap_clk	3.00	2.422	0.38

▢ **Latency (clock cycles)**

▢ **Summary**

Latency		Interval		Type
min	max	min	max	
1155	1155	1155	1155	none

▢ **Detail**

▢ **Instance**

▢ **Loop**

Loop Name	Latency		Iteration Latency	Initiation Interval		Trip Count	Pipelined
	min	max		achieved	target		
- for_shift	512	512	4	-	-	128	no
- for_mac	640	640	5	-	-	128	no

Utilization Estimates

▢ **Summary**

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	1	-	-
Expression	-	-	0	223
FIFO	-	-	-	-
Instance	-	-	-	-
Memory	2	-	0	0
Multiplexer	-	-	-	101
Register	-	-	147	-
Total	2	1	147	324
Available	270	240	84400	42200
Utilization (%)	~0	~0	~0	~0

- Nagyobb órajel frekvencia → több regiszter szint → nagyobb ciklus iteration latency
- Teljes ciklus latency továbbra is: iteráció szám * iteráció latency, mivel nincs pipeline működés

3. Módosítsa az együttható adattípust 32 bitesre (előjeles, 31 bit törtrész), és adaptálja ehhez a többi változó szélességét is. Mit tapasztal szintézis után?

```
typedef ap_fixed<32, 1, AP_RND_CONV, AP_WRAP> coeff_t;
typedef ap_fixed<24, 1, AP_RND_CONV, AP_SAT> din_t;
typedef ap_fixed<32+24, 2, AP_TRN, AP_WRAP> mul_t;
typedef ap_fixed<32+24+7, 2+7, AP_TRN, AP_WRAP> accu_t;
typedef ap_fixed<24, 1, AP_TRN, AP_SAT> dout_t;
```

Performance Estimates

Timing (ns)

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	3.00	1.999	0.38

Latency (clock cycles)

Summary

Latency		Interval		Type
min	max	min	max	
1795	1795	1795	1795	none

Detail

Instance

Loop

Loop Name	Latency		Iteration	Latency	Initiation Interval		Trip Count	Pipelined
	min	max			achieved	target		
- for_shift	512	512	4	4	-	-	128	no
- for_mac	1280	1280	10	10	-	-	128	no

Utilization Estimates

Summary

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	-	0	289
FIFO	-	-	-	-
Instance	-	4	165	22
Memory	2	-	0	0
Multiplexer	-	-	-	133
Register	-	-	271	-
Total	2	4	436	444
Available	270	240	84400	42200
Utilization (%)	~0	1	~0	1

Detail

Instance

Instance	Module	BRAM_18K	DSP48E	FF	LUT
fir_hw_mul_29s_24bkb_U1	fir_hw_mul_29s_24bkb	0	4	165	22
Total	1	0	4	165	22

- 24x32 bites szorzás → több fizikai DSP blokk (szorzás szempontjából 2 elég lenne...) → több regiszter szint → nagyobb ciklus iteration latency
- Megjegyzés. Implementáció (szintézis, place&route) után az erőforrás igény megfelel a várakozásainknak:

```
==== Post-Implementation Resource usage ===
SLICE:      82
LUT:       148
FF:        328
DSP:        2
BRAM:       2
SRL:        1
```

4. Válassza ki a for_shift ciklust, majd adja hozzá a forrás file-hoz az UNROLL direktívát (teljes unroll, így a factor-t nem kell megadni). Mennyiben változik az implementáció eredménye (késleltetés, memória igény, regiszter igény)?

Performance Estimates

- Timing (ns)
 - Summary

Clock	Target	Estimated	Uncertainty
ap_clk	3.00	1.950	0.38
 - Latency (clock cycles)
 - Summary

Latency		Interval		Type
min	max	min	max	
1409	1409	1409	1409	none
 - Detail
 - Instance
 - Loop

Loop Name	Latency		Iteration Latency	Initiation Interval		Trip Count	Pipelined
	min	max		achieved	target		
-for_mac	1280	1280	10	-	-	128	no

Utilization Estimates

- Summary

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	-	0	239
FIFO	-	-	-	-
Instance	-	4	165	22
Memory	3	-	0	0
Multiplexer	-	-	-	1836
Register	-	-	3297	-
Total	3	4	3462	2097
Available	270	240	84400	42200
Utilization (%)	1	1	4	4

Shift unroll: eltűnik a ciklus, de a tömb továbbra is BRAM-ban van → nem oldható meg egy órajel alatt. Ekkor kiolvassa a teljes tömböt regiszterekbe (regiszter szám megnőtt), majd a regiszterekből visszaírja. Olvasás ~64 órajel (128 elem, 2 port), írás szintúgy, így a teljes latency kb a MAC latency + ~128: 1280+128=1408

5. Jelölje ki a mintatár tömbjét a Directive ablakban, majd adja hozzá az ARRAY_PARTITION direktívát „complete” opcióval. Mit tapasztal szintézis után?

Performance Estimates

⊖ **Timing (ns)**

⊖ **Summary**

Clock	Target	Estimated	Uncertainty
ap_clk	3.00	2.584	0.38

⊖ **Latency (clock cycles)**

⊖ **Summary**

Latency		Interval		Type
min	max	min	max	
1282	1282	1282	1282	none

⊖ **Detail**

⊖ **Instance**

⊖ **Loop**

Loop Name	Latency		Iteration	Latency	Initiation Interval		Trip Count	Pipelined
	min	max			achieved	target		
-for_mac	1280	1280		10	-	-	128	no

Utilization Estimates

⊖ **Summary**

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	-	0	239
FIFO	-	-	-	-
Instance	-	4	165	22
Memory	1	-	0	0
Multiplexer	-	-	-	619
Register	-	-	6320	-
Total	1	4	6485	880
Available	270	240	84400	42200
Utilization (%)	~0	1	7	2

- PARTITION szétszedi a tömböt 128 regiszterre → shiftelés egy órajel
- Ez majd LUT shift regiszter lesz, de az erőforrás táblázatban regiszterként jelenik meg
- 1 BRAM az együtttható tömbnek

6. Jelölje ki a for_mac ciklust, s adja meg a PIPELINE direktívát 1-es Iteration Interval használatával. Vizsgálja meg a szintézis eredményét.

Performance Estimates

- Timing (ns)
 - Summary

Clock	Target	Estimated	Uncertainty
ap_clk	3.00	2.584	0.38
 - Latency (clock cycles)
 - Summary

Latency		Interval		Type
min	max	min	max	
139	139	139	139	none
 - Detail
 - Instance
 - Loop

Loop Name	Latency		Iteration Latency	Initiation Interval		Trip Count	Pipelined
	min	max		achieved	target		
-for_mac	136	136	10	1	1	128	yes

Utilization Estimates

- Summary

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	-	0	257
FIFO	-	-	-	-
Instance	-	4	165	22
Memory	1	-	0	0
Multiplexer	-	-	-	644
Register	0	-	6373	32
Total	1	4	6538	955
Available	270	240	84400	42200
Utilization (%)	~0	1	7	2

- PIPELINE direktíva hatása: minden órajelben elkezdődik egy új MAC ciklus iteráció → latency kb.: iteráció szám + iteration latency
- Erőforrás kb. ugyanaz, mint az előző esetben
- Ez már egész jó verzió, 139 órajelenként tud új mintát fogadni, az órajel pedig 333 MHz. Így olyan esetekben, amikor a bemeneti adat mintavételi frekvenciája kisebb, mint 333 MHz/139 = ~2,4 MHz, ez megfelelő feldolgozási sebességű HW.

7. Állítsa át az órajel periódusidőt 2.0 ns-ra, majd fordítsa le a tervet. Mit tapasztal?

Performance Estimates

- Timing (ns)**
 - Summary**

Clock	Target	Estimated	Uncertainty
ap_clk	2.00	1.750	0.25
 - Latency (clock cycles)**
 - Summary**

Latency		Interval		Type
min	max	min	max	
267	267	267	267	none
 - Detail**
 - Instance**
 - Loop**

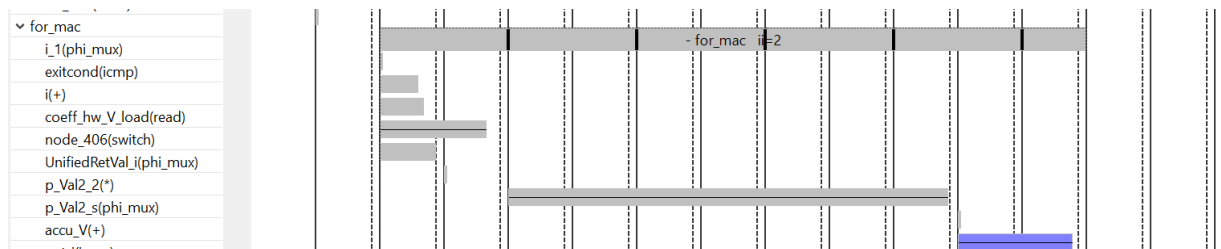
Loop Name	Latency		Iteration Latency	Initiation Interval		Trip Count	Pipelined
	min	max		achieved	target		
- for_mac	264	264	11	2	1	128	yes

Utilization Estimates

- Summary**

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	-	0	191
FIFO	-	-	-	-
Instance	-	4	437	85
Memory	1	-	0	0
Multiplexer	-	-	-	623
Register	0	-	6378	32
Total	1	4	6815	931
Available	270	240	84400	42200
Utilization (%)	~0	1	8	2

- 500 MHz-es órajel esetén a MAC művelet visszacsatolt összeadása (akkumulálás) nem fér bele egy órajelbe, így nem tud $II=1$ megvalósítást generálni. Két órajel periódus lesz egy összeadás, így $II=2$



8. Állítsa vissza az órajel periódusidejét 3 ns-ra. Jelölje ki a for_mac ciklust, s adja meg az UNROLL direktívát 2-es factor használatával. Vizsgálja meg a szintézis eredményét.

Performance Estimates

⊖ **Timing (ns)**

⊖ **Summary**

Clock	Target	Estimated	Uncertainty
ap_clk	3.00	2.519	0.38

⊖ **Latency (clock cycles)**

⊖ **Summary**

Latency		Interval		Type
min	max	min	max	
76	76	76	76	none

⊖ **Detail**

⊖ **Instance**

⊖ **Loop**

Loop Name	Latency		Iteration	Latency	Initiation Interval		Trip Count	Pipelined
	min	max			achieved	target		
- for_mac	73	73		11	1	1	64	yes

Utilization Estimates

⊖ **Summary**

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	-	0	318
FIFO	-	-	-	-
Instance	-	8	330	1390
Memory	1	-	0	0
Multiplexer	-	-	-	63
Register	0	-	6556	32
Total	1	8	6886	1803
Available	270	240	84400	42200
Utilization (%)	~0	3	8	4

- UNROLL=2 → 2 MAC órajelenként → 2x annyi DSP blokk (még mindig rosszul kijelezve...)
- Egy ciklus késleltetése nő, hiszen a két részsorozatot össze kell adni (DSP48 kaszkádosítás)
- Továbbra is 1 BRAM, mert ebből órajelenként 2 együtttható olvasható, a mintatár pedig regiszterekben van, ebből is kinyerhető 2 adat
- A teljes latency 139-ről 73-ra változik, az ütemezési overhead-ek miatt nem feleződik!

9. Módosítsa az UNROLL factor értékét 4-re, s ismételje meg az implementációt. Mit tapasztal?

Performance Estimates

⊖ **Timing (ns)**

⊖ **Summary**

Clock	Target	Estimated	Uncertainty
ap_clk	3.00	2.519	0.38

⊖ **Latency (clock cycles)**

⊖ **Summary**

Latency		Interval		Type
min	max	min	max	
45	45	45	45	none

⊖ **Detail**

⊖ **Instance**

⊖ **Loop**

Loop Name	Latency		Iteration	Latency	Initiation Interval		Trip Count	Pipelined
	min	max			achieved	target		
- for_mac	42	42	12	1	1	32	yes	

Utilization Estimates

⊖ **Summary**

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	-	0	453
FIFO	-	-	-	-
Instance	-	16	660	2780
Memory	4	-	0	0
Multiplexer	-	-	-	63
Register	0	-	6922	32
Total	4	16	7582	3328
Available	270	240	84400	42200
Utilization (%)	1	6	8	7

- 4 MAC órajelenként → 4x annyi DSP blokk
- 1 BRAM-ból nem olvasható ki 4 adat órajelenként → automatikusan particionálja az együtttható tömböt. Valamiért nem 2 dual-port BRAM-ba, hanem 4-be (???)
- Az overhead miatt továbbra sincs 4x gyorsulás az 1 DSP-s verzióhoz képest

10. Állítsa be a for_mac ciklusra a teljes UNROLL-t, és vizsgálja meg az implementáció eredményét.

Performance Estimates

- ▣ **Timing (ns)**
 - ▣ **Summary**

Clock	Target	Estimated	Uncertainty
ap_clk	3.00	2.996	0.38
 - ▣ **Latency (clock cycles)**
 - ▣ **Summary**

Latency		Interval		Type
min	max	min	max	
14	14	14	14	none
 - ▣ **Detail**
 - ▣ **Instance**
 - ▣ **Loop**

N/A

Utilization Estimates

- ▣ **Summary**

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	-	0	7559
FIFO	-	-	-	-
Instance	-	392	21886	1430
Memory	-	-	-	-
Multiplexer	-	-	-	65
Register	-	-	16000	-
Total	0	392	37886	9054
Available	270	240	84400	42200
Utilization (%)	0	163	44	21

- Teljesen párhuzamos megvalósítás → kb annyi 4x DSP blokk, ahány iterációja a MAC ciklusnak van.
- Nincs BRAM felhasználás → együtttható tömb konstansként szerepel (ha írható lenne, akkor regiszterekké particionálná)
- Ennyi erőforrással elvben 1 órajel alatt elvégezhető a feladat, de az ütemezés overhead (start/stop, pipeline latency) miatt sokkal lassabb → nagyon pazarló, ÉRTELMETLEN!

11. Adja meg a PIPELINE direktívát a fir_hw függvényre II=1 paraméterrel. Mennyiben változik az implementáció utáni eredmény?

Performance Estimates

- ▣ **Timing (ns)**
 - ▣ **Summary**

Clock	Target	Estimated	Uncertainty
ap_clk	3.00	2.996	0.38
- ▣ **Latency (clock cycles)**
 - ▣ **Summary**

Latency		Interval		Type
min	max	min	max	
14	14	1	1	function
 - ▣ **Detail**
 - ▣ **Instance**
 - ▣ **Loop**

N/A

Utilization Estimates

- ▣ **Summary**

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	-	0	7571
FIFO	-	-	-	-
Instance	-	392	21886	1430
Memory	-	-	-	-
Multiplexer	-	-	-	-
Register	2	-	17009	1
Total	2	392	38895	9002
Available	270	240	84400	42200
Utilization (%)	~0	163	46	21

- A teljes (top level) függvény pipeline-osítva → minden ciklus unroll-olva → erőforrás igény kb ugyanaz, mint az előbb, latency szintén
- Nincs viszont ütemezési overhead, „órajelenként hívható a függvény”: Interval=1. Ez egy tökéletes, teljesen párhuzamos megvalósítás.

Felmerülő ötlet: részben párhuzamosított verziókhöz használjuk a top level függvény pipeline-osítását magasabb II-vel. Ez önmagában nem elegendő, mert a pipeline direktíva miatt teljesen unroll-olja a MAC ciklust, és hiába nagyobb az II egynél, nem vonja össze egy hardver elembe a műveleteket (sajnos, pedig megtehetné). ALLOCATION direktívával limitálva a felhasználható szorzók számát viszont rábírható erre. A 8. pontban 4-es UNROLL-t használtunk, ami azt jelenti, hogy teoretikusan 128/4=32 órajel alatt a MAC műveletek elvégezhetőek, tehát elméletileg lehetséges PIPELINE II=32, MAC=4 kombináció előállítás. Ehhez az alábbi direktívák szükségesek a fir_hw függvényre:

```
#pragma HLS ALLOCATION instances=mul limit=4 operation
#pragma HLS PIPELINE II=32
```

Ennek eredménye:

Performance Estimates

Timing (ns)

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	3.00	3.150	0.38

Latency (clock cycles)

Summary

Latency		Interval		Type
min	max	min	max	
43	43	32	32	function

Detail

Instance

Loop

Utilization Estimates

Summary

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	-	0	6482
FIFO	-	-	-	-
Instance	-	24	1152	92
Memory	-	-	-	-
Multiplexer	-	-	-	1201
Register	-	-	10848	-
Total	0	24	12000	7775
Available	270	240	84400	42200
Utilization (%)	0	10	14	18

Ez már jó megoldás.

12. Térjen vissza 6. pont direktíváihoz, és módosítsa az adattípusokat úgy, hogy mind a bemeneti- és kimeneti minták, mind pedig az együtthatók 18 bites, előjeles, 17 bit törtrészt tartalmazó számok legyenek. Megfelel az implementáció eredménye a várakozásoknak?

```
typedef ap_fixed<18, 1, AP_RND_CONV, AP_WRAP> coeff_t;
typedef ap_fixed<18, 1, AP_RND_CONV, AP_SAT> din_t;
typedef ap_fixed<18+18, 2, AP_TRN, AP_WRAP> mul_t;
typedef ap_fixed<18+18+7, 2+7, AP_TRN, AP_WRAP> accu_t;
typedef ap_fixed<18, 1, AP_TRN, AP_SAT> dout_t;
```

Performance Estimates

Timing (ns)

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	3.00	2.584	0.38

Latency (clock cycles)

Summary

Latency		Interval		Type
min	max	min	max	
261	261	261	261	none

Detail

Instance

Loop

Loop Name	Latency		Iteration	Latency	Initiation Interval		Trip Count	Pipelined
	min	max			achieved	target		
- for_mac	258	258	5	5	2	1	128	yes

Utilization Estimates

Summary

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	1	-	-
Expression	-	-	0	173
FIFO	-	-	-	-
Instance	-	-	-	-
Memory	1	-	0	0
Multiplexer	-	-	-	623
Register	-	-	4657	-
Total	1	1	4657	796
Available	270	240	84400	42200
Utilization (%)	~0	~0	5	1

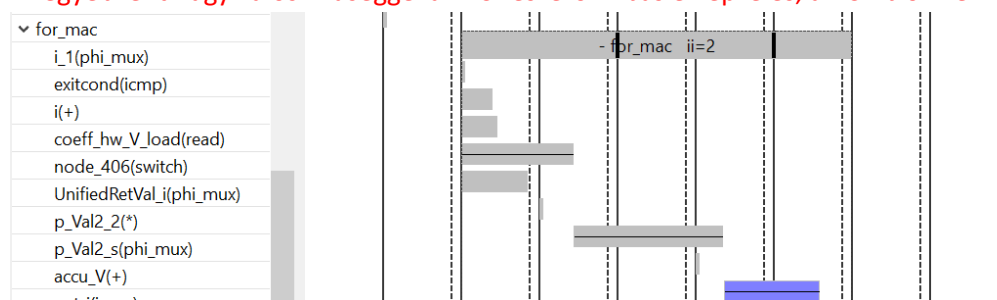
Detail

Instance

DSP48

Instance	Module	Expression
fir_hw_mac_muladdbkb_U1	fir_hw_mac_muladdbkb	i0 + i1 * i2

- Ugyanazt várnánk, mint a 6. pontban, úgyhogy nem.
- A DSP blokk száma ok (18x18 szorzás), de láthatóan az akkumulálást is a DSP blokkon belül oldja meg (egyébként helyesen), de valamiért úgy dönt, hogy ezt nem tudja elvégezni 1 órajel alatt → ugyanaz az eredménye, mint az előbbieken a 2 ns-os órajelnél.
- Ez egyébként nagy valószínűséggel a HLS részéről hibás elképzelés, a DSP blokk ennél gyorsabb.



- 3.3 ns-os órajellel már megfelelő az eredmény.

Cirkuláris buffer használata

1. Maradva a 18 bites adattípusoknál, írja át a szűrő kódját úgy, hogy a mintatár BRAM-ban megvalósított cirkuláris buffer legyen. Az eddig használt direktívákat távolítsa el! Milyen változótípust célszerű használni a tömbök címzéséhez?

```
typedef ap_uint<7> addr_t;
typedef ap_fixed<18, 1, AP_RND_CONV, AP_WRAP> coeff_t;
typedef ap_fixed<18, 1, AP_RND_CONV, AP_SAT> din_t;
typedef ap_fixed<18+18, 2, AP_TRN, AP_WRAP> mul_t;
typedef ap_fixed<18+18+7, 2+7, AP_TRN, AP_WRAP> accu_t;
typedef ap_fixed<18, 1, AP_TRN, AP_SAT> dout_t;
```

```
void fir_hw(din_t *input, dout_t *res)
{
    static din_t smp1[N];
    static addr_t wr_addr;
    addr_t rd_addr;
    addr_t addr;
    int i;
    accu_t accu;

    addr = wr_addr;
    smp1[addr] = *input;

    accu = 0;
    for_mac: for (i=0; i<N; i++)
    {
        accu = accu + (coeff_hw[i] * smp1[addr]);
        addr--;
    }

    wr_addr++;
    *res = accu;
}
```

2. A MAC for ciklusra állítsa be a PIPELINE direktívát II=1 értékkel. Megfelel az implementáció a várakozásoknak?

Performance Estimates

Timing (ns)

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	3.30	2.950	0.41

Latency (clock cycles)

Summary

Latency		Interval		Type
min	max	min	max	
134	134	134	134	none

Detail

Instance

Loop

Loop Name	Latency		Iteration	Latency	Initiation Interval		Trip Count	Pipelined
	min	max			achieved	target		
- for_mac	131	131		5	1	1	128	yes

Utilization Estimates

Summary

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	1	-	-
Expression	-	-	0	197
FIFO	-	-	-	-
Instance	-	-	-	-
Memory	2	-	0	0
Multiplexer	-	-	-	87
Register	0	-	177	32
Total	2	1	177	316
Available	270	240	84400	42200
Utilization (%)	~0	~0	~0	~0

- Megfelel a várakozásoknak.

3. Mennyire ideális a BRAM használat?

Nem az, a 2 darab 128x18-as tömb bőven elférne 1 BRAM-ban (ami 1024x18 méretű).

4. Állítsa be az ARRAY_MAP direktívát mind a mintatárra, mind pedig az együtthatókészletre. A típus legyen horizontal, az instance pedig ugyanaz (pl. array0) mindkét tömb esetében. Mit tapasztal a szintézis után?

Performance Estimates

▣ **Timing (ns)**

▣ **Summary**

Clock	Target	Estimated	Uncertainty
ap_clk	3.30	2.950	0.41

▣ **Latency (clock cycles)**

▣ **Summary**

Latency		Interval		Type
min	max	min	max	
134	134	134	134	none

▣ **Detail**

▣ **Instance**

▣ **Loop**

Loop Name	Latency		Iteration	Latency	Initiation Interval		Trip Count	Pipelined
	min	max			achieved	target		
- for_mac	131	131		5	1	1	128	yes

Utilization Estimates

▣ **Summary**

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	1	-	-
Expression	-	-	0	197
FIFO	-	-	-	-
Instance	-	-	-	-
Memory	1	-	0	0
Multiplexer	-	-	-	87
Register	0	-	177	32
Total	1	1	177	316
Available	270	240	84400	42200
Utilization (%)	~0	~0	~0	~0

- Várakozásainknak megfelelően a két tömböt egy memóriában valósítja meg. Mivel a BRAM 2 portos, így az egyik porton olvashatók az együtthatók, a másikon a minták.

5. Vizsgálja meg az implementáció eredményét a MAC ciklus részleges 2-es UNROLL beállításával. Mi limitálja az elérhető sebességet?

Performance Estimates

Timing (ns)

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	3.30	2.950	0.41

Latency (clock cycles)

Summary

Latency		Interval		Type
min	max	min	max	
137	137	137	137	none

Detail

Instance

Loop

Loop Name	Latency		Iteration	Latency	Initiation Interval		Trip Count	Pipelined
	min	max			achieved	target		
- for_mac	134	134		9	2	1	64	yes

Utilization Estimates

Summary

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	2	-	-
Expression	-	-	0	265
FIFO	-	-	-	-
Instance	-	-	-	-
Memory	1	-	0	0
Multiplexer	-	-	-	141
Register	0	-	302	32
Total	1	2	302	438
Available	270	240	84400	42200
Utilization (%)	~0	~0	~0	1

- Órajelenként 4 adatra lenne szüksége a 2 MAC egységnek, ami 1 BRAM-ból nem olvasható ki, így eredményül II=2 megoldás generálódik, ami semmivel sem gyorsabb, mint az előző, csak több erőforrást igényel → memória sávszélesség limitált.

```

WARNING: [SCHED 204-69] Unable to schedule 'load' operation ('array@load_2', fir_hw.cpp:282) on array 'array0' due to limited memory ports. Please consider using a memory core with more ports or partitioning the array 'array0'.
INFO: [SCHED 204-61] Pipelining result : Target II = 1, Final II = 2, Depth = 9.
WARNING: [SCHED 204-21] Estimated clock period (2.95ns) exceeds the target (target clock period: 3.3ns, clock uncertainty: 0.4125ns, effective delay budget: 2.8875ns).
WARNING: [SCHED 204-21] The critical path consists of the following:
  *mul' operation ('p_Val12_3_1', fir_hw.cpp:282) (0 ns)
  *add' operation ('tmp_7', fir_hw.cpp:282) (2.95 ns)

```

6. Megoldást jelenthet, hogy a memóriák összevonása előtt megduplázzuk a szószélességet, majd az így keletkező tömböket vonjuk össze (így 2 porton 36-36 bit érhető el, azaz 4x18). Ehhez mind az együtttható, mind pedig a mintatár tömbre állítsunk be ARRAY_RESHAPE direktívát 2-es factor és cyclic mód használatával, majd az ARRAY_MAP direktívával vonjuk őket össze. Mi lesz a szintézis eredménye?

Performance Estimates

▢ **Timing (ns)**

▢ **Summary**

Clock	Target	Estimated	Uncertainty
ap_clk	3.30	2.950	0.41

▢ **Latency (clock cycles)**

▢ **Summary**

Latency		Interval		Type
min	max	min	max	
140	140	140	140	none

▢ **Detail**

▢ **Instance**

▢ **Loop**

Loop Name	Latency		Iteration	Latency	Initiation Interval		Trip Count	Pipelined
	min	max			achieved	target		
-for_mac	134	134		9	2	1	64	yes

Utilization Estimates

▢ **Summary**

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	2	-	-
Expression	-	-	0	914
FIFO	-	-	-	-
Instance	-	-	-	-
Memory	2	-	0	0
Multiplexer	-	-	-	143
Register	0	-	548	64
Total	2	2	548	1121
Available	270	240	84400	42200
Utilization (%)	~0	~0	~0	2

▢ **Detail**

▢ **Instance**

▢ **DSP48**

▢ **Memory**

Memory	Module	BRAM_18K	FF	LUT	Words	Bits	Banks	W*Bits*Banks
array0_U	fir_hw_array0	2	0	0	128	36	1	4608
Total		1	2	0	128	36	1	4608

Bár az elvi memória port szám megfelelő, a megoldás nem jó. A 2-es unroll miatt pl. az első iterációban a legújabb és az egyvel régebbi mintára lenne szükség. Minden második bemeneti mintánál ez a két 18 bites szó egyetlen 36 bites szóban lesz, viszont a maradék esetekben a két 18 bites érték nem ugyanabban a 36 bites szóban van, így hiába 36 bit széles a memória, nem olvashatók ki egyetlen órajel alatt (két 36 bites szóból lenne szükség 18-18 bitre).

Ráadásul a HLS 1 helyett 2 BRAM-t használ.

7. Elrontva a szűrő funkcionalitását a MAC for ciklusában módosítsa a mintatár címzését úgy, hogy az is a ciklusváltozóval történjen. Így mi lesz a szintézis eredménye? Magyarázható a 6. pontban mutatott viselkedés?

Performance Estimates

⊟ **Timing (ns)**

⊟ **Summary**

Clock	Target	Estimated	Uncertainty
ap_clk	3.30	2.950	0.41

⊟ **Latency (clock cycles)**

⊟ **Summary**

Latency		Interval		Type
min	max	min	max	
76	76	76	76	none

⊟ **Detail**

⊟ **Instance**

⊟ **Loop**

Loop Name	Latency		Iteration	Initiation Interval		Trip Count	Pipelined
	min	max		achieved	target		
- for_mac	70	70	8	1	1	64	yes

Utilization Estimates

⊟ **Summary**

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	2	-	-
Expression	-	-	0	848
FIFO	-	-	-	-
Instance	-	-	-	-
Memory	2	-	0	0
Multiplexer	-	-	-	98
Register	0	-	546	96
Total	2	2	546	1042
Available	270	240	84400	42200
Utilization (%)	~0	~0	~0	2

⊟ **Detail**

⊟ **Instance**

⊟ **DSP48**

⊟ **Memory**

Memory	Module	BRAM_18K	FF	LUT	Words	Bits	Banks	W*Bits*Banks
array0_U	fir_hw_array0	2	0	0	128	36	1	4608
Total		1	2	0	128	36	1	4608

A fenti olvasási problémát a szűrő elrontása kiküszöböli, ekkor mindig egy 36 bites szó két 18 bites részére van szükség, így ez a megoldás implementálható („csak” funkcionálisan helytelen). És a HLS egy helyett 2 BRAM-t használ, feleslegesen.