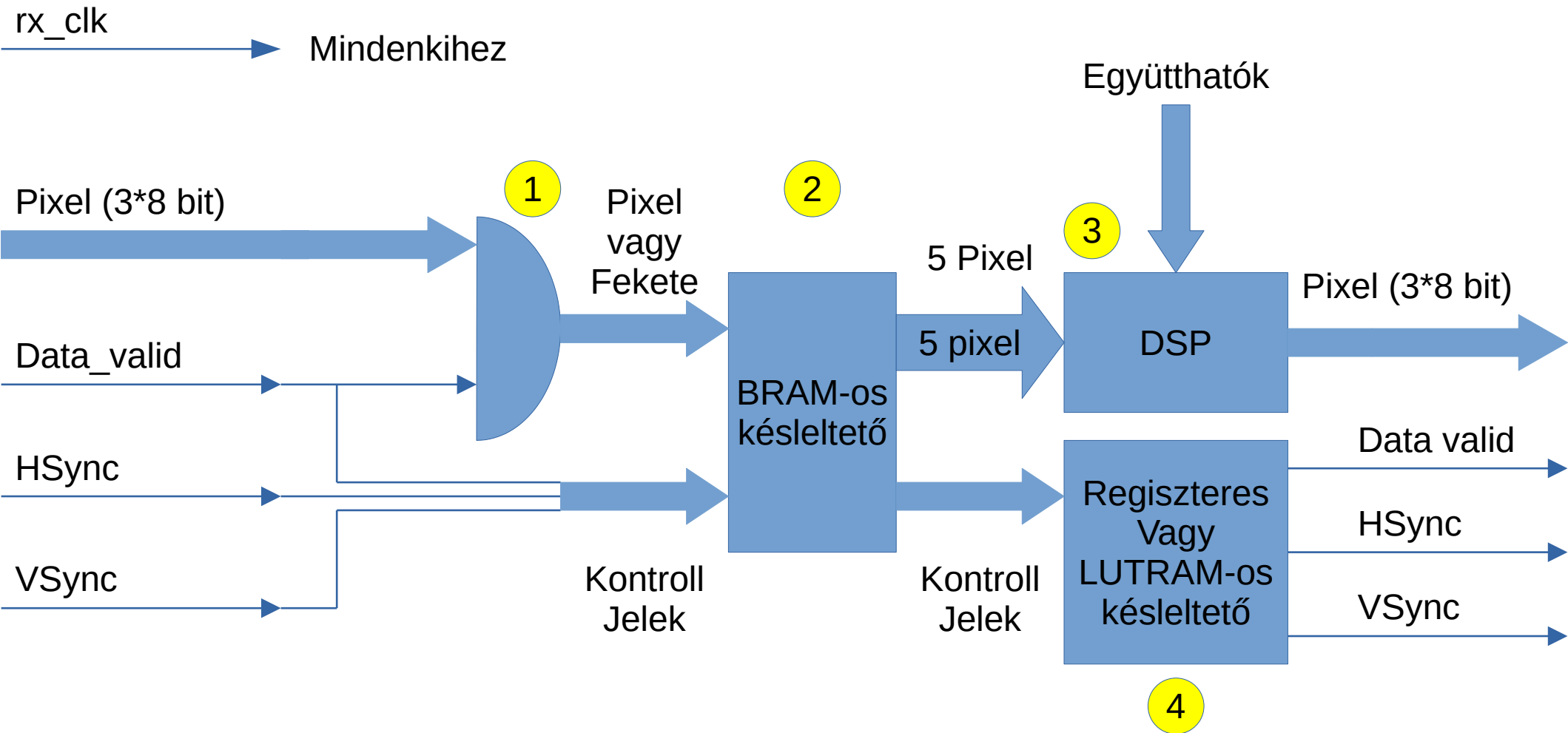


Logikai tervezés – Házi feladat blokkvázlat
Bálint Gergely
Szilágyi Gábor

Teljes blokkvázlat:

bemenetek

kimenetek



1. Blokk részletesen: sok ÉS kapu, a bejövő Data_valid jelet a bejövő Pixel bitjeivel egyenként összeésseli, így fekete pixellel számolunk tovább, ha a bejövő pixel érték érvénytelen.

Esetleg kell még az elejére egy pipeline regiszter-réteg, mert a data valid jel fanoutja itt 25 (26 a továbbvitt, késleltetett példányával együtt)

2. Blokk részletesen: beérkező pixel mentése és 5 különböző késleltetésű változat (pa, ..., pe) előállítása a DSP rész bemenetéhez, valamint színcsatornánként 1-1 kontroll jel késleltetése a blokkramok 9. bitjében

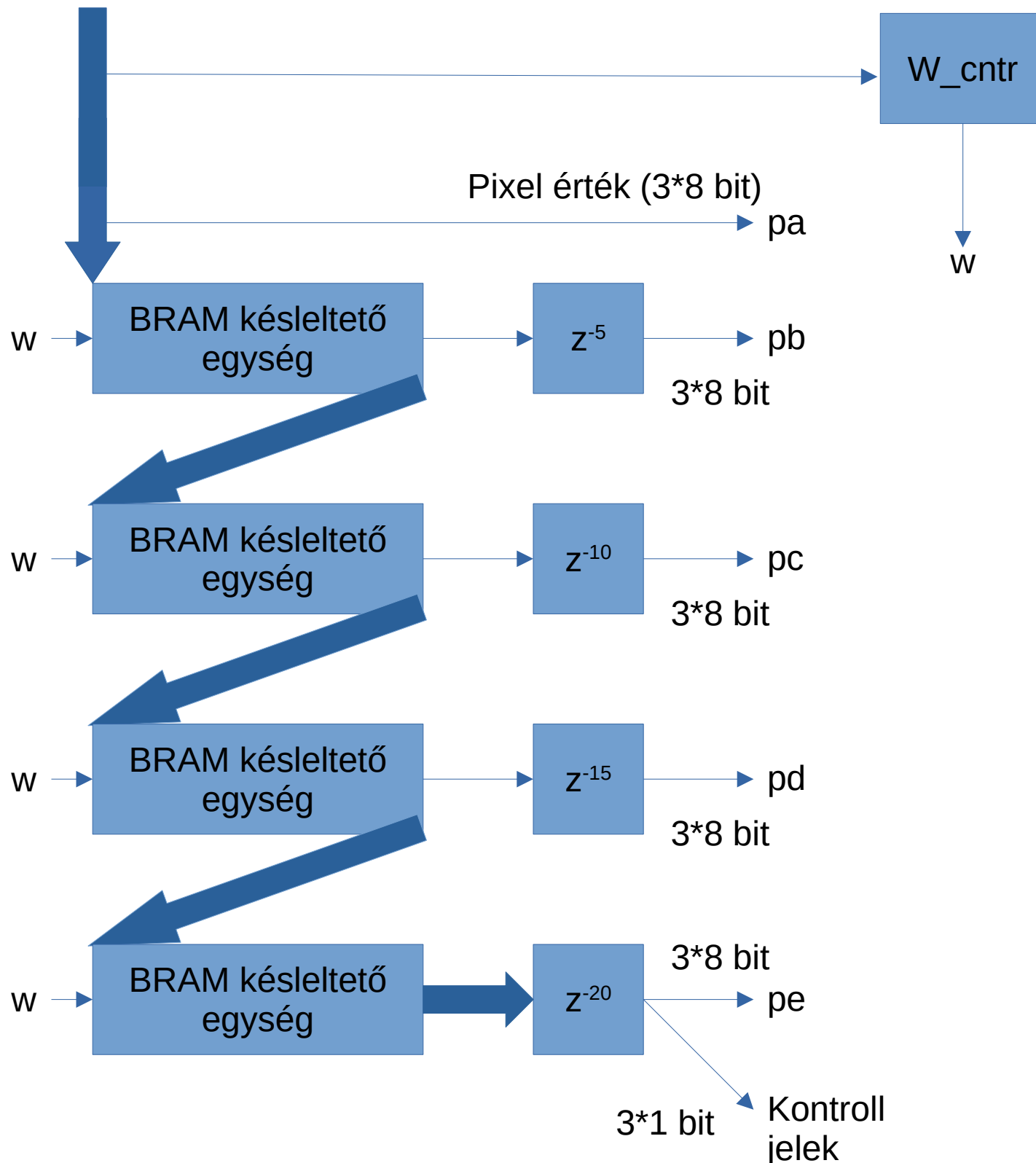
z^{-n}

n órajellel késleltető
FF-halom, vagy
BRAM

BRAM késleltető
egység

3 db párhuzamos, 4k x 9 bit
blokkram, ciklikus címzés single
port, read first mód, 8 bit (1
színcsatorna) + 1 bit
(data_valid/hsync/vsync egy-egy
színcsatorna mellé)

Pixel + Kontroll: $3 \cdot (8+1) = 27$ bit

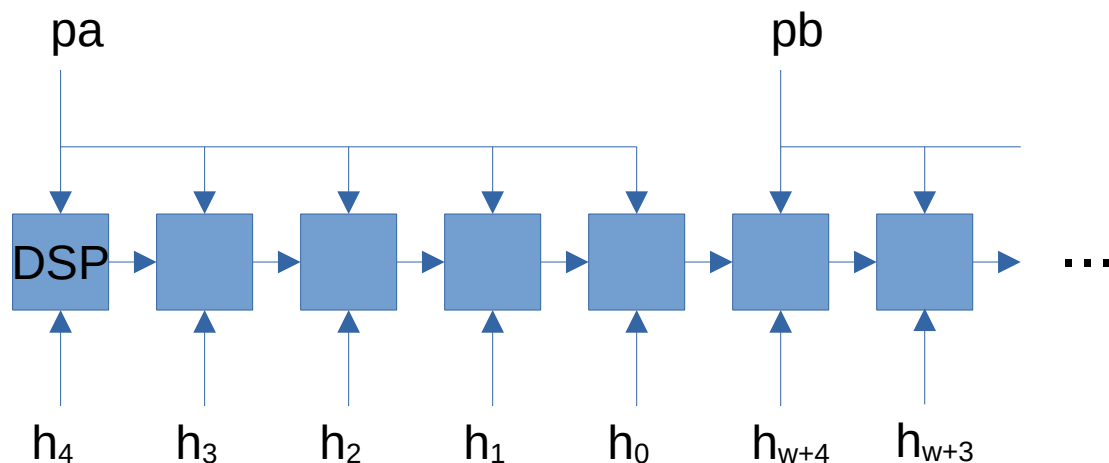


Egy számláló+ regiszter, ami
megszámolja a két Hsync
felfutó él között eltelt
órajelciklusokat, vagyis a kép
szélességét az invalid
pixelekkal együtt (ez a w, ami
12 bites, mert 1080p esetén
ennyibe fér bele a kép
szélessége). Az előállított w
értéket használják a BRAM-
ok a ciklikus címzés
reszetteléséhez.

Így minden sor hosszát az
előző sor hosszának
feltételezünk. Működés
közbeni felbontásváltás
esetén ez nem probléma,
mert a hiba csak egy sorig áll
fent, ami szabad szemmel
nem látható.

3. Blokk részletesen: DSP

Szorzás: a súlyozómátrix sorait meg lehet valósítani kaszkád struktúrával, mert az csak ugyanannak a pixel streamnek a különböző késleltetésű változatait adja össze.



Egy DSP blokk elvileg 3 órajel késleltetést jelent (ha minden belső pipeline regiszterjét használjuk) VERIFIKÁLNI

1 új pixelérték kiszámításához a 25 együtthatót 25 különböző idővel késleltetett pixelértékekkel kell szorozni, és ezeket összeadni (késleltetések és együtthatók a következő oldalon). Az együtthatóablak egy sorában lévő pixelek késleltetése egyesével növekszik, ez azt jelenti, hogy az a sor elején lévő pixelt az aktuális és következő 4 órajelben kell felhasználnunk, majd egy sorhossznyi késleltetés múlva újra.

Fent látható, hogy az adott pixelt a kaszkádosított szorzó-összeadó blokkba emiatt egyszerre 5 helyen csatoljuk be, a +4 késleltetésről a számítás gondoskodik.

A pb pixelt 5 hellyel később csatoljuk be a kaszkádszámításba, így elcsúszna a hozzá tartozó pa pixeltől. Ennek a megoldására az előző oldalon látható 5-ös késleltetések szolgálnak. Ezekkel a később becsatolt pixelek késleltetését "kiegyenlítjük".

Késleltetések a
konvolúcióhoz:

4+4W	3+4W	2+4W	1+4W	0+4W
4+3W	3+3W	2+3W	1+3W	0+3W
4+2W	3+2W	2+2W	1+2W	0+2W
4+W	3+W	2+W	1+W	0+W
4	3	2	1	0

Együtthatók nevei:

h_{4W+4}	h_{4W+3}	h_{4W+2}	h_{4W+1}	h_{4W+0}
h_{3W+4}	h_{3W+3}	h_{3W+2}	h_{3W+1}	h_{3W+0}
h_{2W+4}	h_{2W+3}	h_{2W+2}	h_{2W+1}	h_{2W+0}
h_{W+4}	h_{W+3}	h_{W+2}	h_{W+1}	h_{W+0}
h_4	h_3	h_2	h_1	h_0

4. Blokk részletesen: Státusz jelek késleltetése

A státuszjeleket a pixelekkel együtt késleltetjük a második blokk végéig. Itt a pixelek a DSP kaszkádba kerülnek, így a státuszjeleket a DSP kaszkád késleltetésével megfelelő mértékben kell késleltetni. Ez BRAM-mal és LUT-tal is megvalósítható, mivel viszonylag rövid késleltetésről van szó.