

Distributed FPGA for enhanced Image Processing

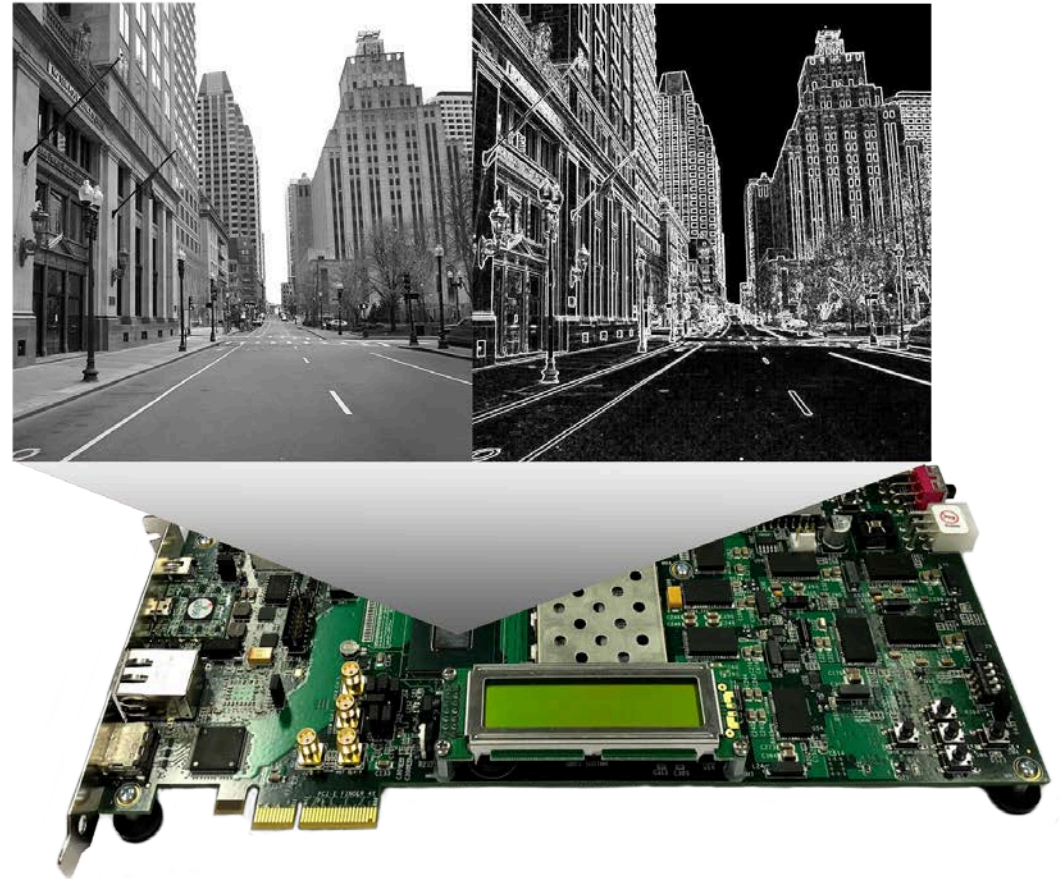
Bildverarbeitung mit Vivado HLS

Inhaltsverzeichnis

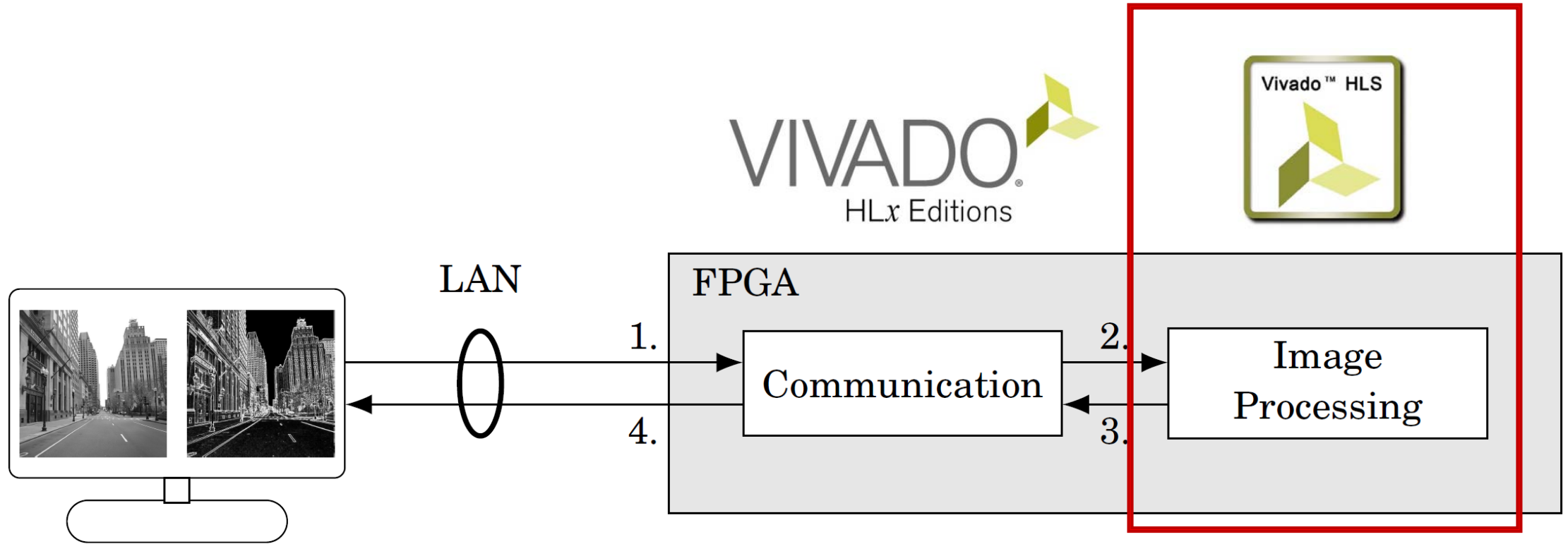
- Projekt
- Vivado HLS Überblick
- Datenverarbeitung mit Vivado HLS
- Fazit

Ausgangslage

- 1500 MP Bilder
- Preprocessing
- Distributed FPGA Lösung



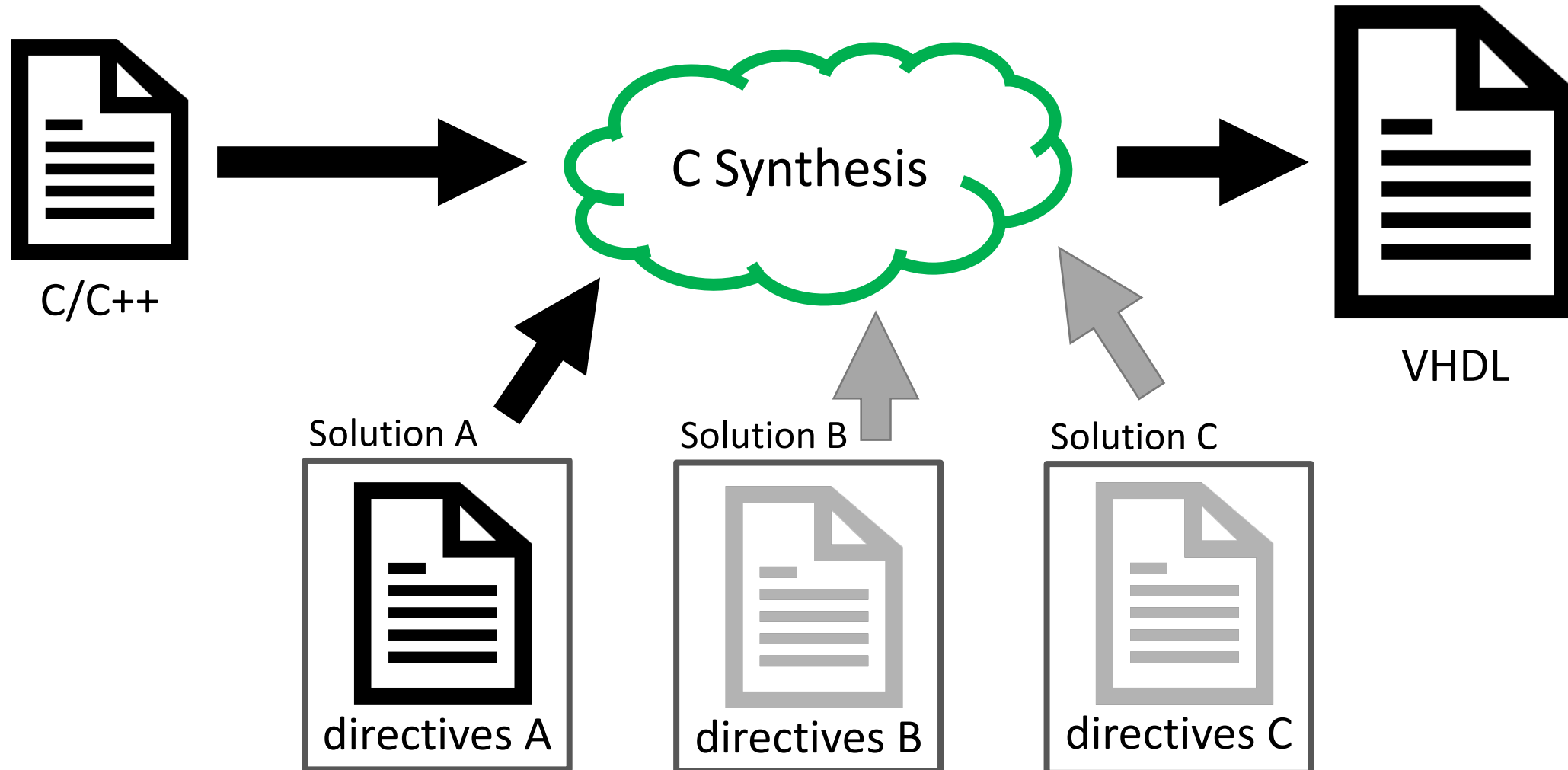
Aufbau



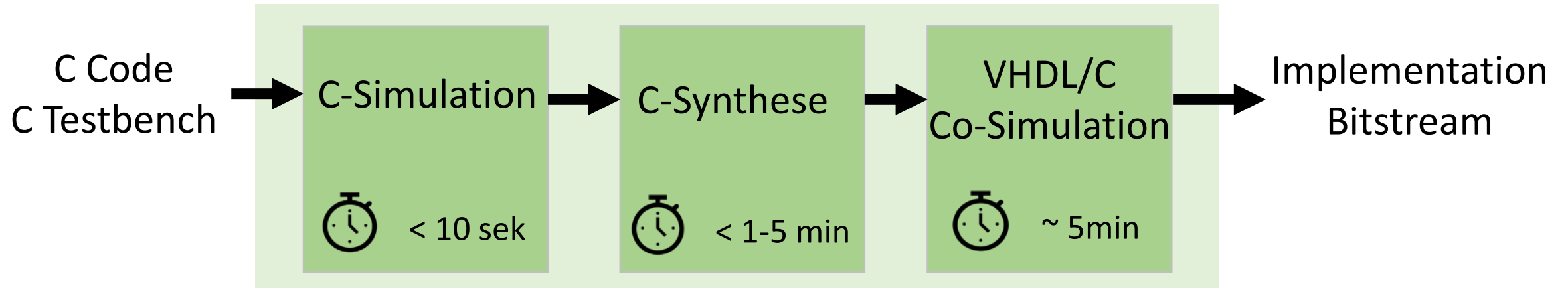
Vivado HLS: Übersicht

- Konzept: C/C++ als Sprache für Hardwarebeschreibung
- Design Flow
- Erweiterungen in C/C++
- GUI
- Pragmas

Konzept



Design Flow

**C-Simulation**

Schneller Bearbeiten/Kompilieren/Testen Kreislauf

C-Synthese

C Code wird in VHDL Code übersetzt

VHDL/C Co-Simulation

VHDL Code wird simuliert und die Resultate mit der C-Simulation verglichen

Erweiterungen in C/C++

Arbiträre Datentypen

```
ap_uint<12>
```

C++ Klassen

```
hls::stream
```

Directives (Pragmas)

Deren Verständnis ist der Schlüssel zum Erfolg mit Vivado HLS

Vivado HLS 17.3 (Home/Noah/Git/Diip/FPGA/HLS/Build/ECC)

File Edit Project Solution Window Help

Debug Synthesis Analysis

tb_ecc.cpp ecc.cpp

```
15
16
17
18 void ecc(AXI_STREAM_8 &inData, AXI_STREAM_8 &outMean, AXI_STREAM_14 &outVar) {
19
20     // *****
21     // Variables
22     AXI_VALUE_8 inPixel;
23     AXI_VALUE_8 out_mean;
24     AXI_VALUE_14 out_var;
25
26     static apuint19_t sum_Pixel;
27     static apuint8_t n_Mean;
28     static apuint14_t n_Var;
29     static apuint8_t pixel[WIN_SIZE];
30     static apuint8_t tmp_Pixel[WIN_SIZE];
31
32
33     // *****
34     // Initialization
35     // *****
36     // Read data and calculate mean
37     sum_Pixel = 0;
38     loop_rdata:for(uint16_t i = 0; i < WIN_SIZE; i++) {
```

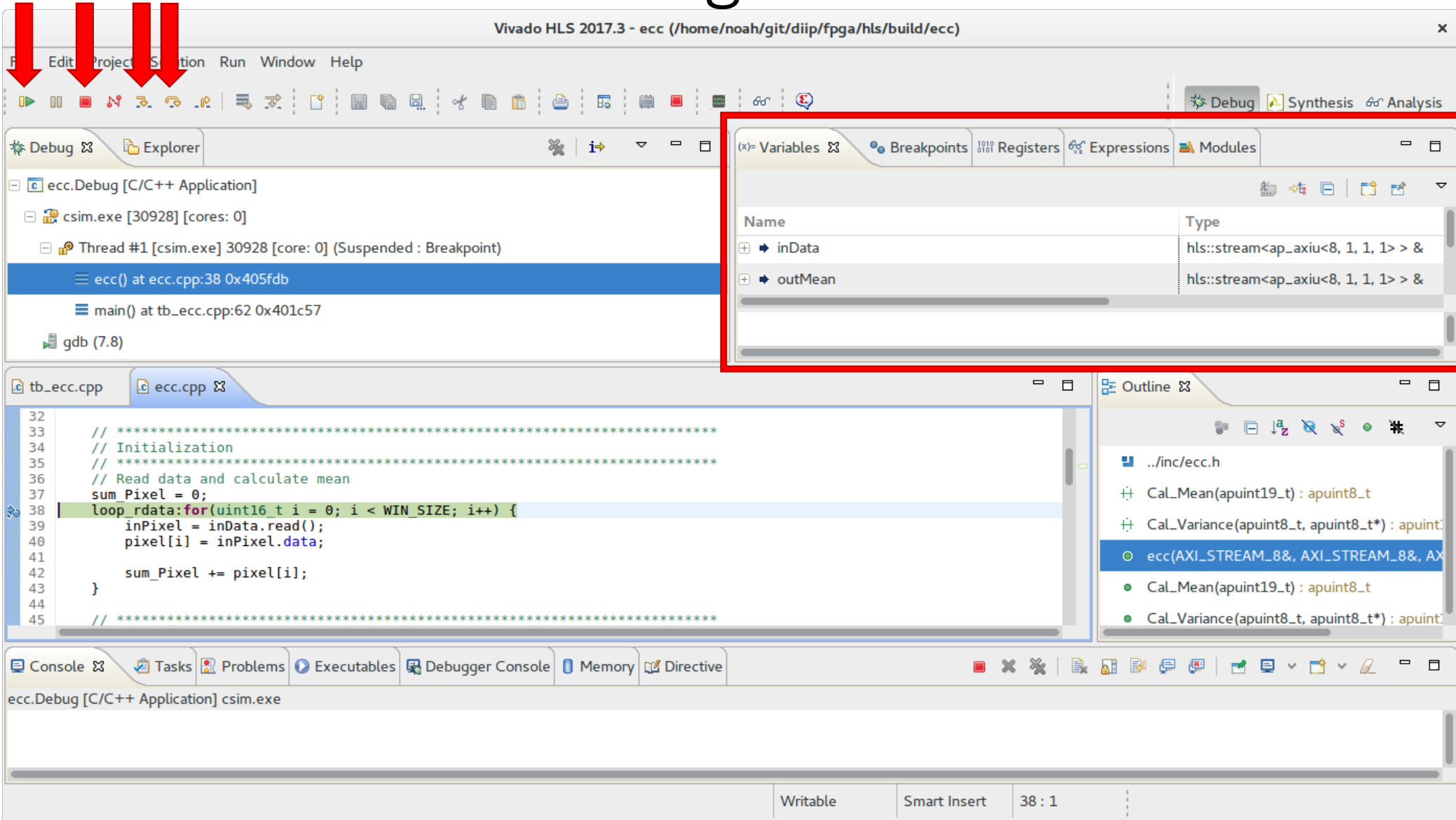
Outline Directive

- Ca_Mean
- Ca_Variance
- ecc
 - % HLS INTERFACE ap_ctrl_hs port=return
 - inData
 - % HLS INTERFACE axis register reverse port=inData
 - outMean
 - % HLS INTERFACE axis register forward port=outMe
 - outVar
 - % HLS INTERFACE axis register forward port=outVar
 - inPixel
 - out_mean

Console Errors Warnings Debugger Console

<terminated> ecc.Debug [C/C++ Application] csim.exe

Writable Smart Insert 39 : 30



Vivado HLS 2017.3 - ecc (/home/noah/git/diip/fpga/hls/build/ecc)

File Edit Project Settings Run Window Help

Debug Explorer

ecc.Debug [C/C++ Application]

- csim.exe [30928] [cores: 0]
 - Thread #1 [csim.exe] 30928 [core: 0] (Suspended : Breakpoint)
 - ecc() at ecc.cpp:38 0x405fdb
 - main() at tb_ecc.cpp:62 0x401c57

gdb (7.8)

tb_ecc.cpp ecc.cpp

```
32
33 // *****
34 // Initialization
35 // *****
36 // Read data and calculate mean
37 sum_Pixel = 0;
38 loop_rdata:for(uint16_t i = 0; i < WIN_SIZE; i++) {
39     inPixel = inData.read();
40     pixel[i] = inPixel.data;
41     sum_Pixel += pixel[i];
42 }
43
44
45 // *****
```

Variables

Name	Type
inData	hls::stream<ap_axiu<8, 1, 1, 1> &
outMean	hls::stream<ap_axiu<8, 1, 1, 1> &

Outline

- ../inc/ecc.h
- Cal_Mean(apuint19_t) : apuint8_t
- Cal_Variance(apuint8_t, apuint8_t*) : apuint8_t
- ecc(AXI_STREAM_8&, AXI_STREAM_8&, AXI_STREAM_8&) : ecc
- Cal_Mean(apuint19_t) : apuint8_t
- Cal_Variance(apuint8_t, apuint8_t*) : apuint8_t

Console

ecc.Debug [C/C++ Application] csim.exe

Writable Smart Insert 38 : 1

Vivado HLS 2017.3 - ecc (/home/noah/git/diip/fpga/hls/build/ecc)

File Edit Project Solution Window Help

Debug Synthesis Analysis

Explorer tb_ecc.cpp ecc.cpp Synthesis(sol4_unroll) Performance(sol4_unroll) Outline Directive

Timing (ns)

Latency (clock cycles)

Summary

Latency		Interval		
min	max	min	max	Type
473	473	474	474	none

Detail

Utilization Estimates

Summary

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	2	-	-
Expression	-	1	0	518
FIFO	-	-	-	-
Instance	0	3	592	2251
Memory	42	-	0	0
Multiplexer	-	-	-	2357
Register	-	-	318	-
Total	42	6	910	5126
Available	730	740	269200	129000
Utilization (%)	5	~0	~0	3

Console Errors Warnings Debugger Console

Vivado HLS Console
INFO: [HLS 200-111] Elapsed time: 0.74 seconds, current allocated memory: 117.055 MB.
INFO: [RTMG 210-278] Implementing memory 'ecc pixel V 0 ram (RAM 2P BRAM)' using block RAMs with power-on initialization.

Vivado HLS 2017.3 - ecc (/home/noah/git/diip/fpga/hls/build/ecc)

File Edit Project Solution Window Help

Debug Synthesis Analysis

Module Hierarchy

	BRAM	DSP	FF	LUT	Latency	Interval	Pipeline type
ecc	42	6	910	5126	473	474	none
Cal_Variance	0	3	592	2146	26	26	none

tb_ecc.cpp ecc.cpp Synthesis(sol4_unroll) Performance(sol4_unroll)

Current Module : ecc

Operation/Control Step	C0	C1	C2	C3	C4	C5	C6
1-36 loop_rdata		█					
37 mul2(*)			█				
38 node_222(write)			█				
39 node_223(write)			█				
40 node_216(write)				█			
41 Cal_Variance(function)				█	█		
42 node_219(write)					█	█	
43 node_221(write)					█	█	
44 node_218(write)						█	█
45-... loop_while							█

Performance Profile Resource Profile

	Pipelined	Latency	Initiation Interval	Iteration Latency	Trip count
ecc	-	473	474	-	-
loop_rdata	yes	441	1	1	441
loop_while	no	0	-	98	0
loop_strData	yes	20	1	2	20
loop_addData	yes	22	1	3	21
loop_setData	yes	21	1	2	21

Performance Resource

Properties Warnings

Property	Value

Vivado 2017.3

File Edit Tools Window Layout View Run Help Quick Access

10 us

Default Layout

SIMULATION - Simulation Result - ecc.wdb

Scope x Sources

Search: mean (52 matches)

Name

- aparb_ecc_top
 - AESL_inst_ecc
 - AESL_axi_s_inst_inData
 - AESL_axi_s_inst_outMean
 - AESL_axi_s_inst_outVar

Objects

Search: mean (52 matches)

Name

- outMean_TDATA[7:0]
- outMean_TVALID
- outMean_TREADY
- outMean_TKEEP[0:0]
- outMean_TSTRB[0:0]
- outMean_TUSER[0:0]
- outMean_TLAST[0:0]
- outMean_TID[0:0]
- outMean_TDEST[0:0]
- outMean_V_data_V_1_data_in[7:0]
- outMean_V_data_V_1_vld_in
- outMean_V_data_V_1_ack_out
- outMean_V_data_V_1_data_reg[7:0]
- outMean_V_data_V_1_sRdy
- outMean_V_data_V_1_mVld
- outMean_V_data_V_1_areset_d
- outMean_V_keep_V_1_vld_in
- outMean_V_keep_V_1_ack_out
- outMean_V_keep_V_1_sRdy
- outMean_V_keep_V_1_mVld
- outMean_V_keep_V_1_areset_d
- outMean_V_strb_V_1_vld_in
- outMean_V_strb_V_1_ack_out
- outMean_V_strb_V_1_sRdy
- outMean_V_strb_V_1_mVld
- outMean_V_strb_V_1_areset_d
- outMean_V_user_V_1_vld_in

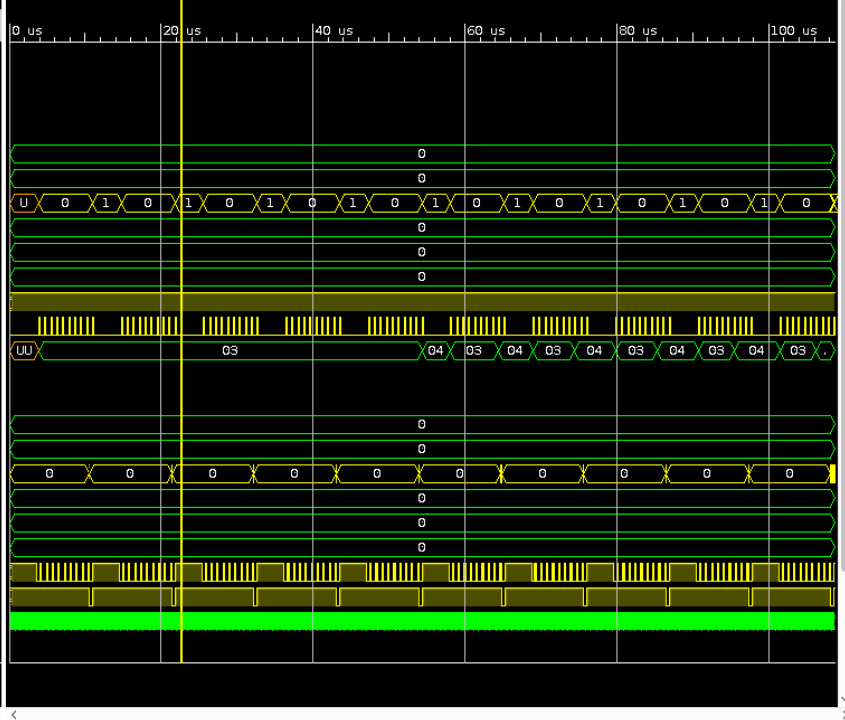
ecc.wcfg*

Search: mean (52 matches)

Name	Value
Design Top Signals	
C Outputs	
outVar(axis)	
outMean(axis)	
outMean_TDEST[0:0]	0
outMean_TID[0:0]	0
outMean_TLAST[0:0]	1
outMean_TUSER[0:0]	0
outMean_TSTRB[0:0]	0
outMean_TKEEP[0:0]	0
outMean_TREADY	1
outMean_TVALID	0
outMean_TDATA[7:0]	03
C Inputs	
inData(axis)	
inData_TDEST[0:0]	0
inData_TID[0:0]	0
inData_TLAST[0:0]	0
inData_TUSER[0:0]	0
inData_TSTRB[0:0]	0
inData_TKEEP[0:0]	0
inData_TREADY	1
inData_TVALID	1
inData_TDATA[7:0]	04
Block-level IO Handshake	

0 us 20 us 40 us 60 us 80 us 100 us

22.766950 us



Tcl Console x Messages

INFO: [IP_Flow 19-1704] No user IP repositories specified
 INFO: [IP_Flow 19-2313] Loaded Vivado IP repository '/opt/Xilinx/Vivado/2017.3/data/ip'.

Type a Tcl command here

Sim Time: 108636 ns

Pragmas

Schnittstellen

- interface
- protocol
- stream

Ressource

- resource
- array_map
- array_partition
- array_reshape
- data_pack
- dependence
- allocation
- reset

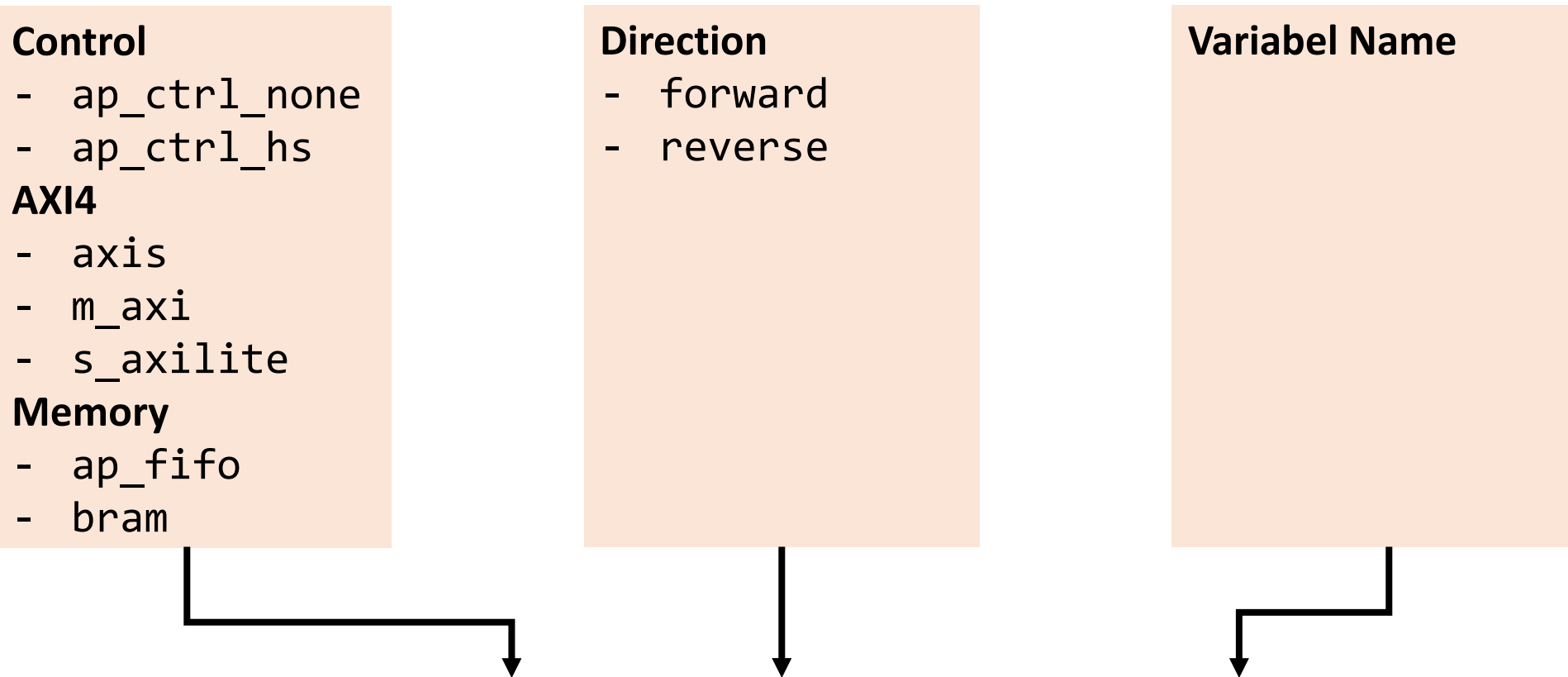
Throughput/Area

- unroll
- pipeline
- inline
- dataflow
- loop_flatten
- loop_merge

Verschiedene

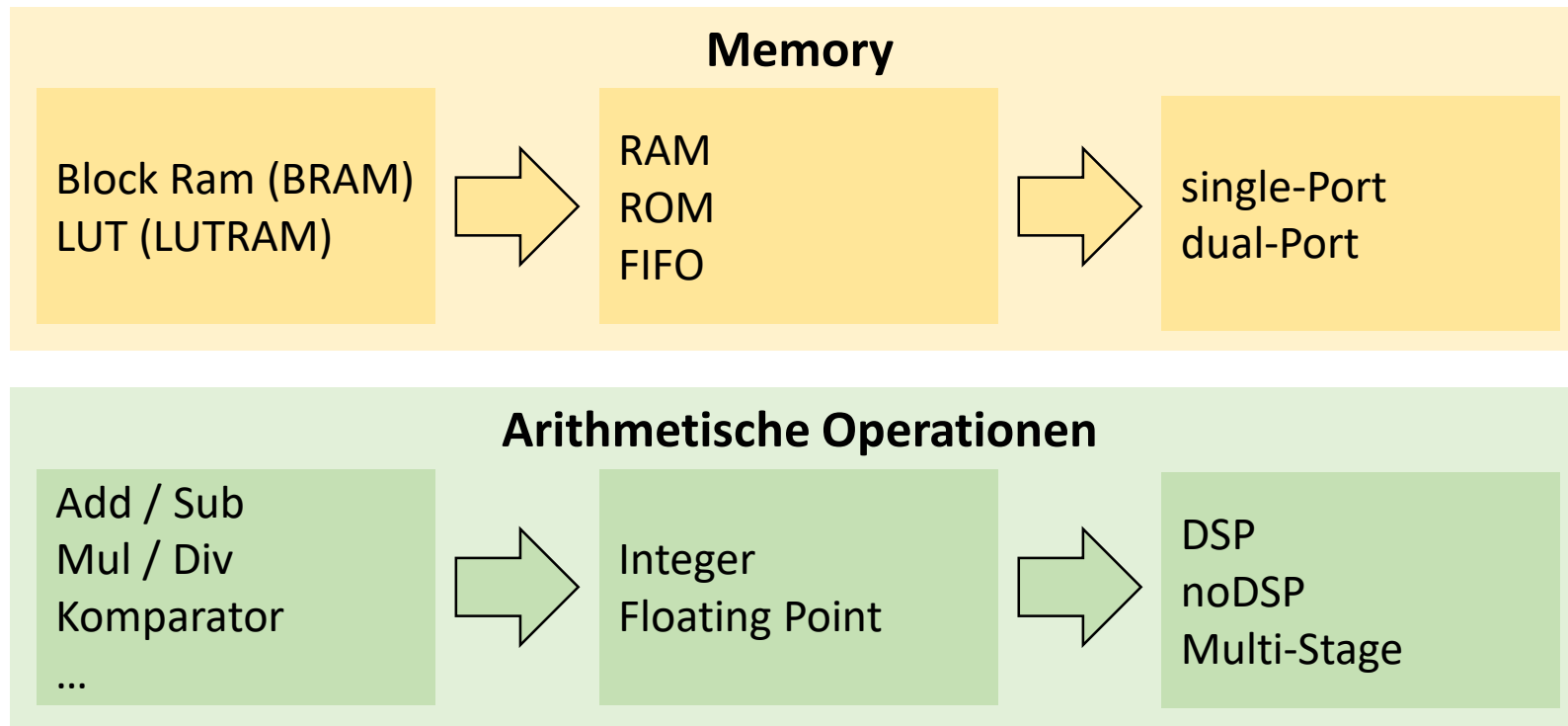
- clock
- loop_tripcount

Pragma: INTERFACE



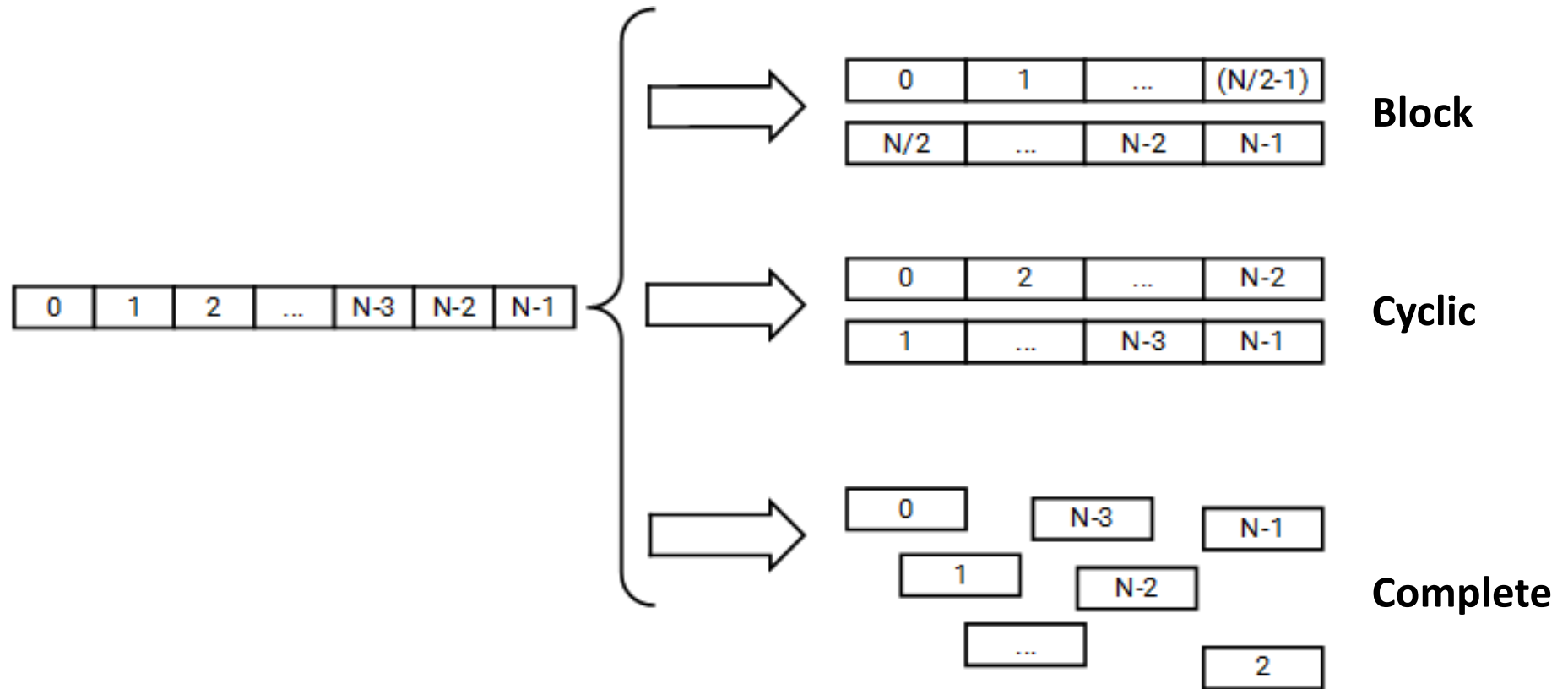
```
#pragma HLS INTERFACE axis register_mode=forward port=outMean
```

Pragma: RESSOURCE



```
#pragma HLS RESOURCE variable=pixel core=RAM_2P_BRAM
```

Pragma: ARRAY_PARTITION



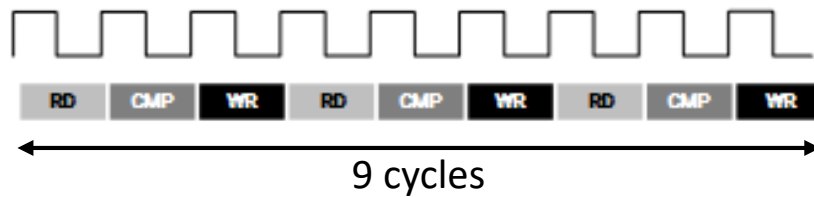
```
#pragma HLS ARRAY_PARTITION variable=pixel dim=1 cyclic factor=21
```

Pragma: PIPELINE / UNROLL

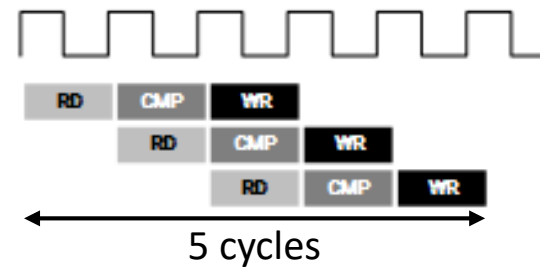
```
void func(m,n,o) {
    for (i=2;i>=0;i--) {
        op_Read;
        op_Compute;
        op_Write;
    }
}
```



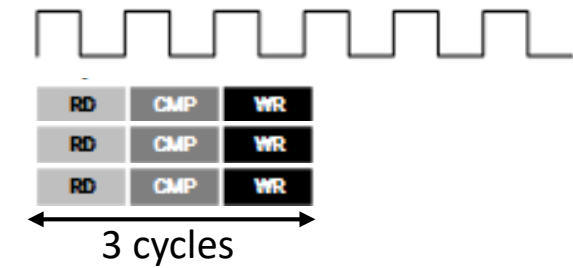
Sequentiell



Pipeline

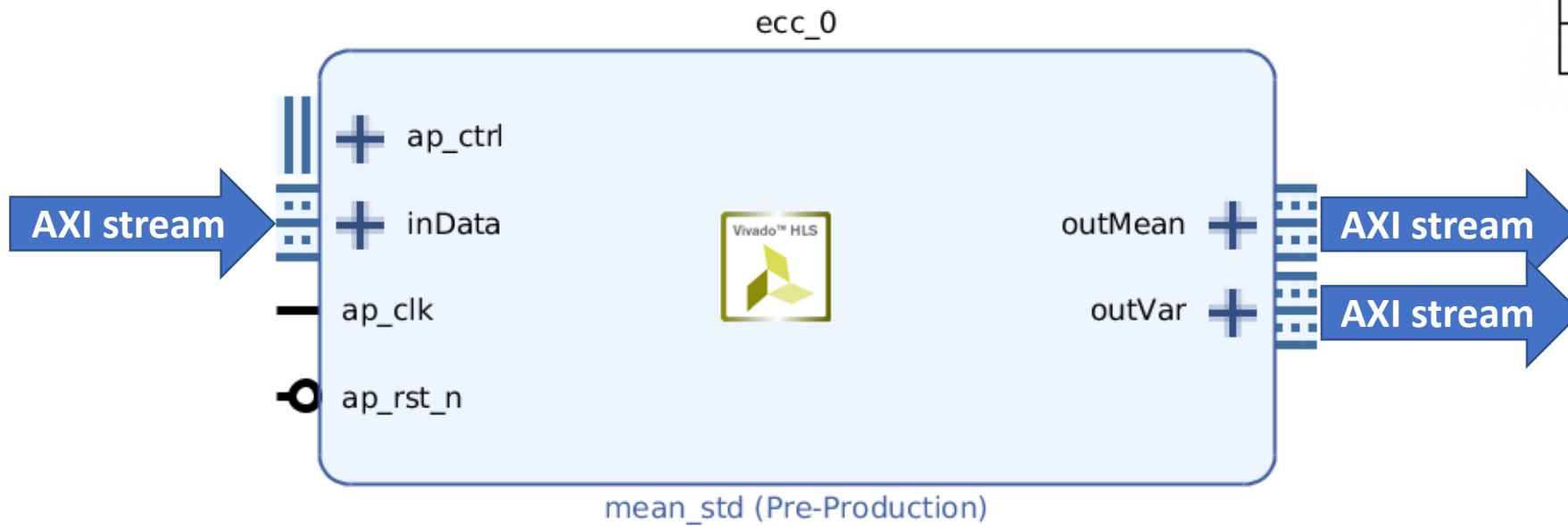


Unroll



#pragma HLS PIPELINE
#pragma HLS UNROLL factor=21

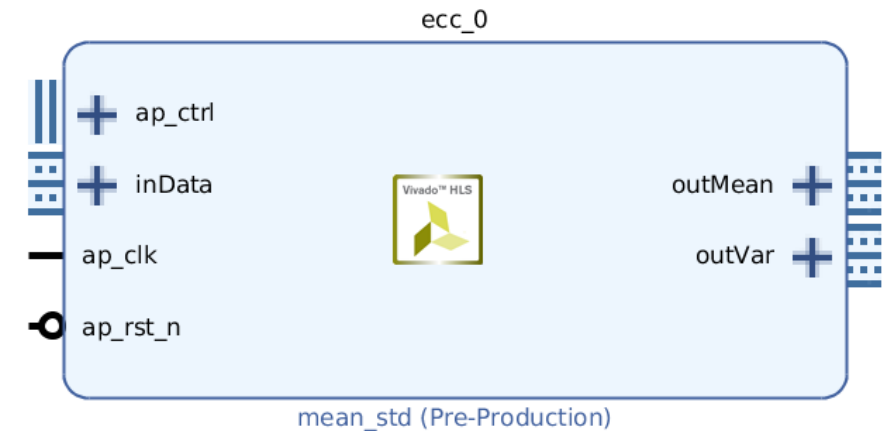
Ausgangslage



Problem

$$\mu = \frac{\sum_{i=1}^N p_i}{N}$$

$$\sigma^2 = \frac{\sum_{i=1}^N (p_i - \mu)^2}{N - 1}$$



Solutions

Interface

- `#pragma HLS INTERFACE axis register reverse port=inDATA`

Array Partition

- `#pragma HLS ARRAY_PARTITION variable=pixel cyclic factor=21 dim=1`

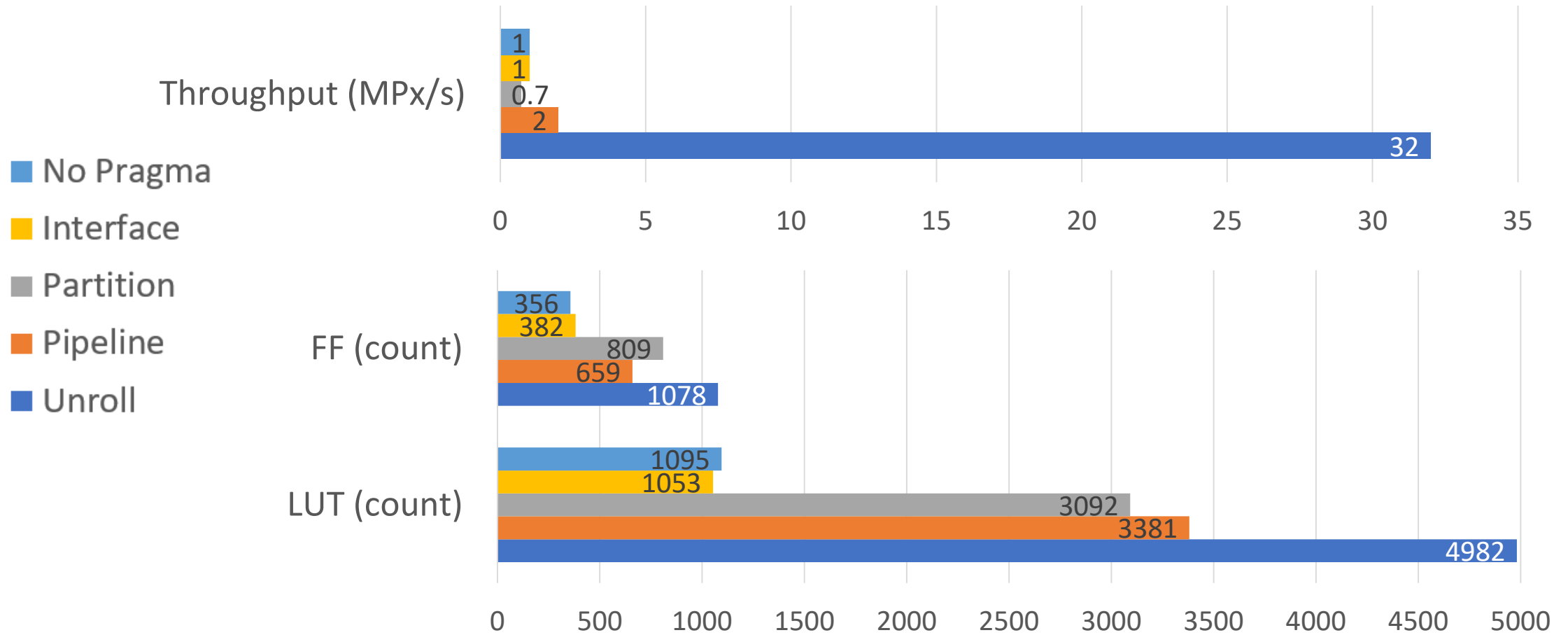
Pipeline Loops

- `#pragma HLS PIPELINE`

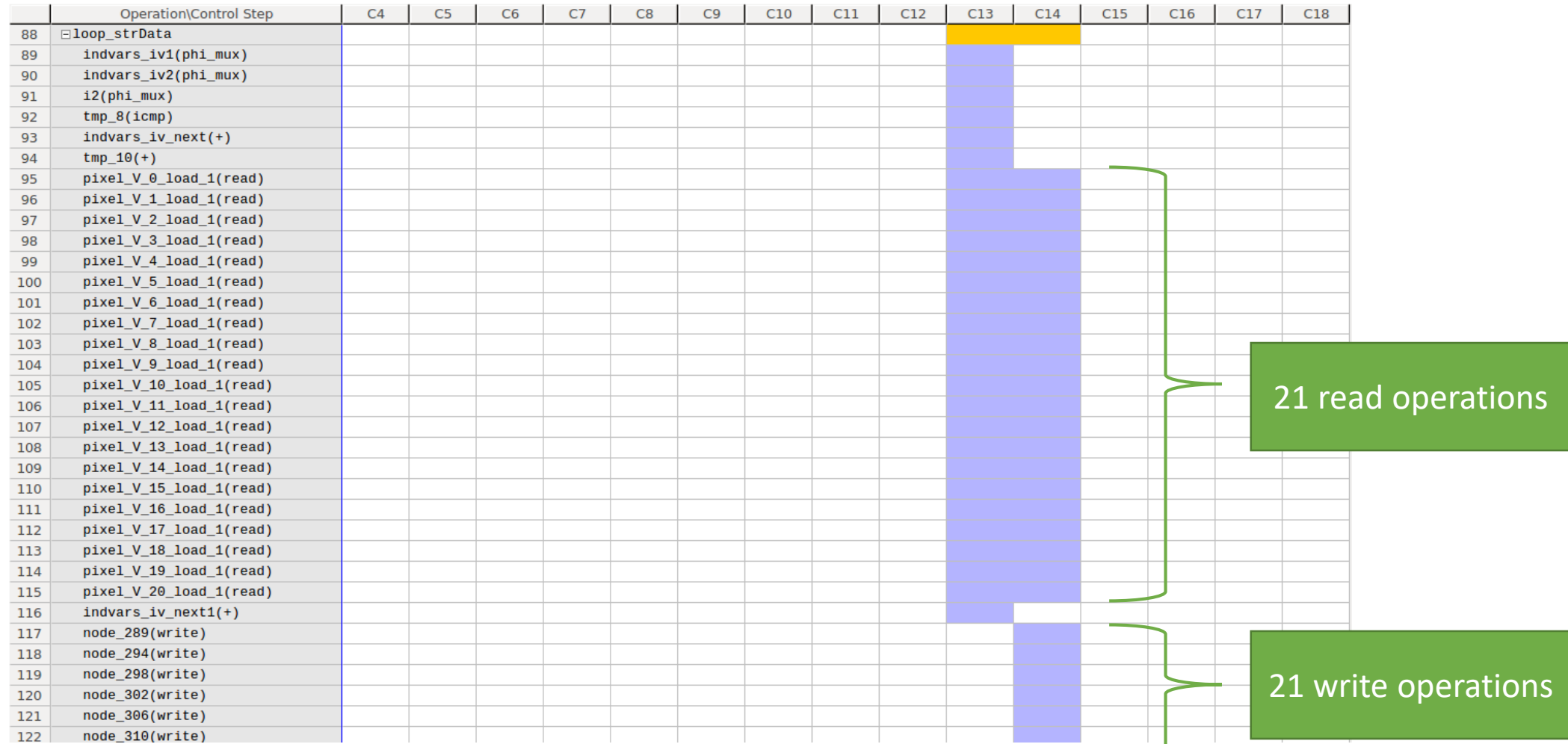
Unroll Loops

- `#pragma HLS UNROLL factor=21`

Vergleich



Analyse



Fazit

Vorteile

- Schneller Design Flow
- Lösungen vergleichen
- Portabel



Nachteile

- Mächtiges Tool das es zu beherrschen gilt
- Nicht nur einfach C/C++ Code

Fazit



Stolpersteine

- Kleine Änderungen können zu grossen Unterschieden in Zeit und Ressourcennutzung führen
 - **Nur ein Pragma auf einmal ändern**
- Hardwarenahes Denken wird vernachlässigt
 - **Flache Hierarchie im C/C++ Code**

Fazit

Unsere Erfahrungen

- Pragmas müssen richtig eingesetzt werden
- Mit wenig VHDL Kenntnissen zum Erfolg
- AXI Interfaces schnell umgesetzt
- Auch FSM möglich

Distributed FPGA for enhanced Image Processing

Bildverarbeitung mit Vivado HLS

Noah Hütter
Jan Stocker

noahhuetter@gmail.com
janstocker@gmx.ch

UG902 Vivado HLS User Guide
UG871 Vivado HLS Tutorial
HLS Pragmas
Präsentation & Code

[PDF](#)
[PDF](#)
[Site](#)
[Site](#)



bit.do/ECC18