UNIVERSITÄT HEIDELBERG

SPADIC
Self triggered Pulse Amplification and Digitization asIC

SPADIC 1.0

Tim Armbruster
tim.armbruster@ziti.uni-heidelberg.de

FEE/DAQ Workshop Mannheim

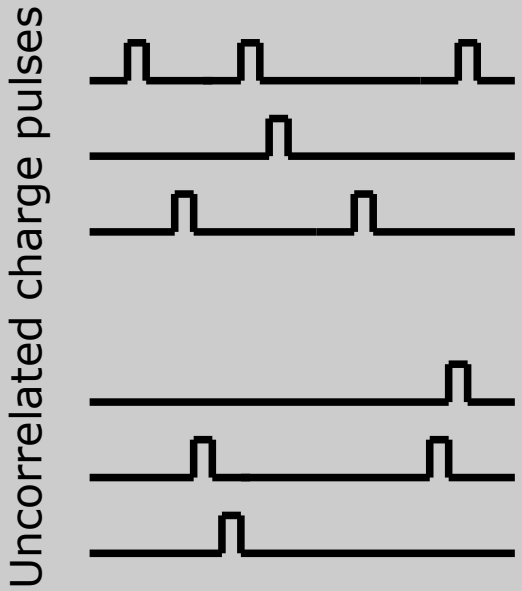
January 2012

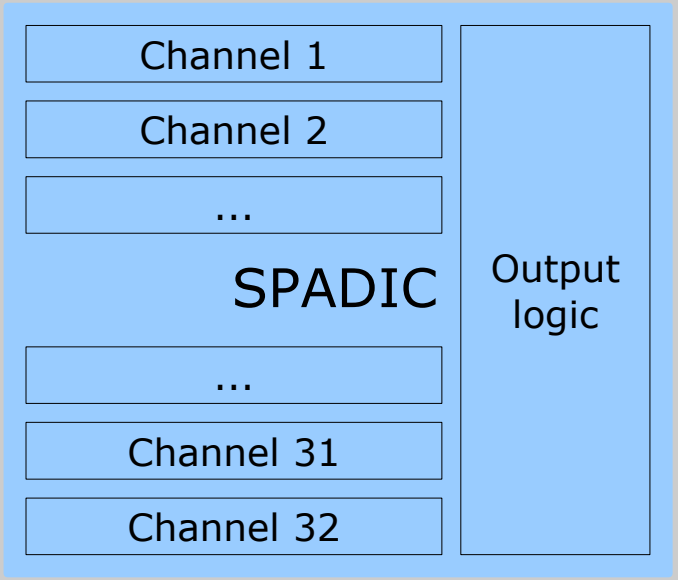
Visit http://www.spadic.uni-hd.de

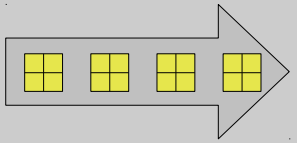
1. SPADIC Architecture

Introduction to SPADIC 1.0

SPADIC: **S**elf-triggered **P**ulse **A**mplification and **D**igitization as**IC**







Output Packages @ 1Gbps (serial)

Abstract Data Flow Concept

Charge Pulse Amplification and Shaping →
Continuous Digitization →
Continuous Filtering →
Digital Hit Detection →
Package Building →
DAQ Protocol Encoding →
Fast Serial Output Interface

See next slides

Possible SPADIC user

- **TRD**
- Maybe RICH
- MUCH ???
- ...

An abstract point of view: The 4 SPADIC Parts

Part 1:
physical adapter – input cell, preamp/shaper, ADC, ... → this talk

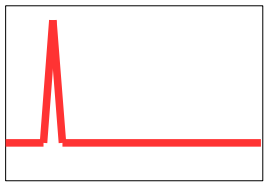
Part 2:
digital processing – IIR filter → Michael Krieger

Part 3:
data extraction – hit detector, package builder, neighbor logic, … → this talk

Part 4:
transfer mechanism – CBM net, LVDS driver → this talk, see also Frank Lemke

The processing chain of SPADIC 1.0 – Step 1

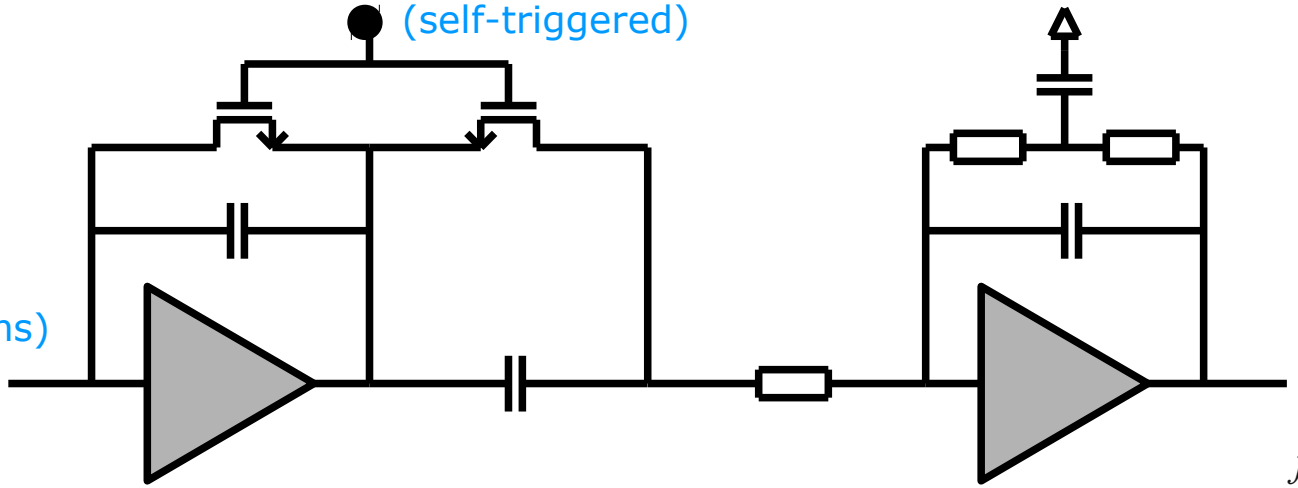
Step 1: Amplification + Shaping



Fast charge pulse from detector (< 10ns)

(both polarities possible)

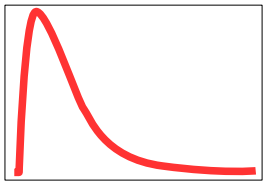
Continuous rest (self-triggered)



Charge Amplifier

Shaper

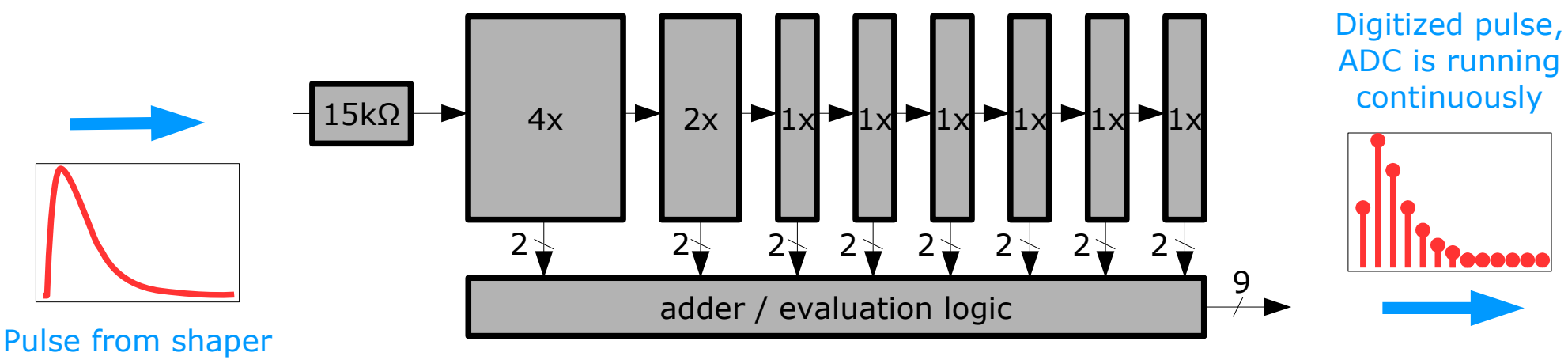
Voltage pulse, shape depends only on shaping-time:



$$f(t) = \left(\frac{t}{T}\right)\exp\left(\frac{t}{T}\right)$$

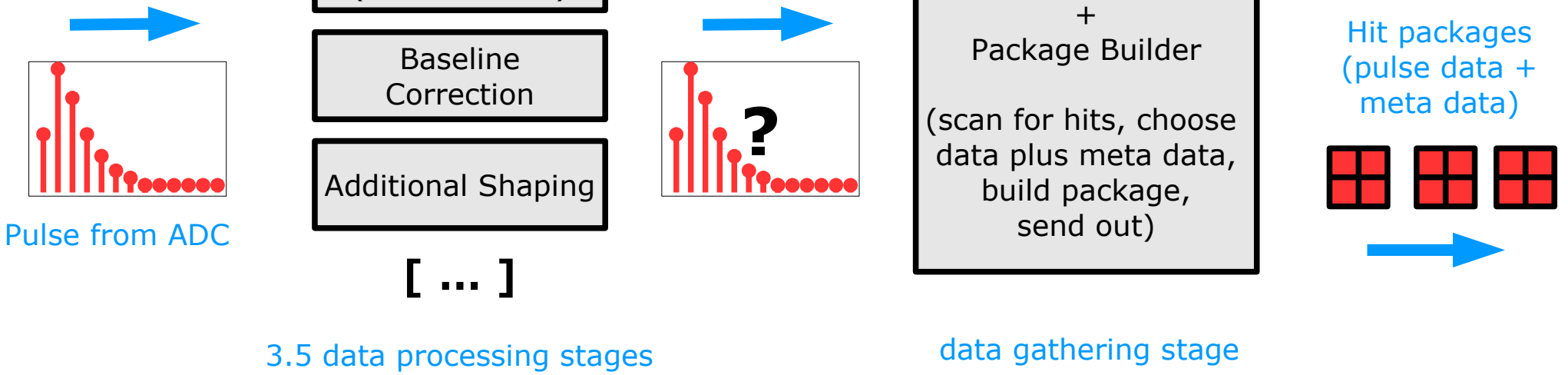
The processing chain of SPADIC 1.0 – Step 2

Step 2: Digitization



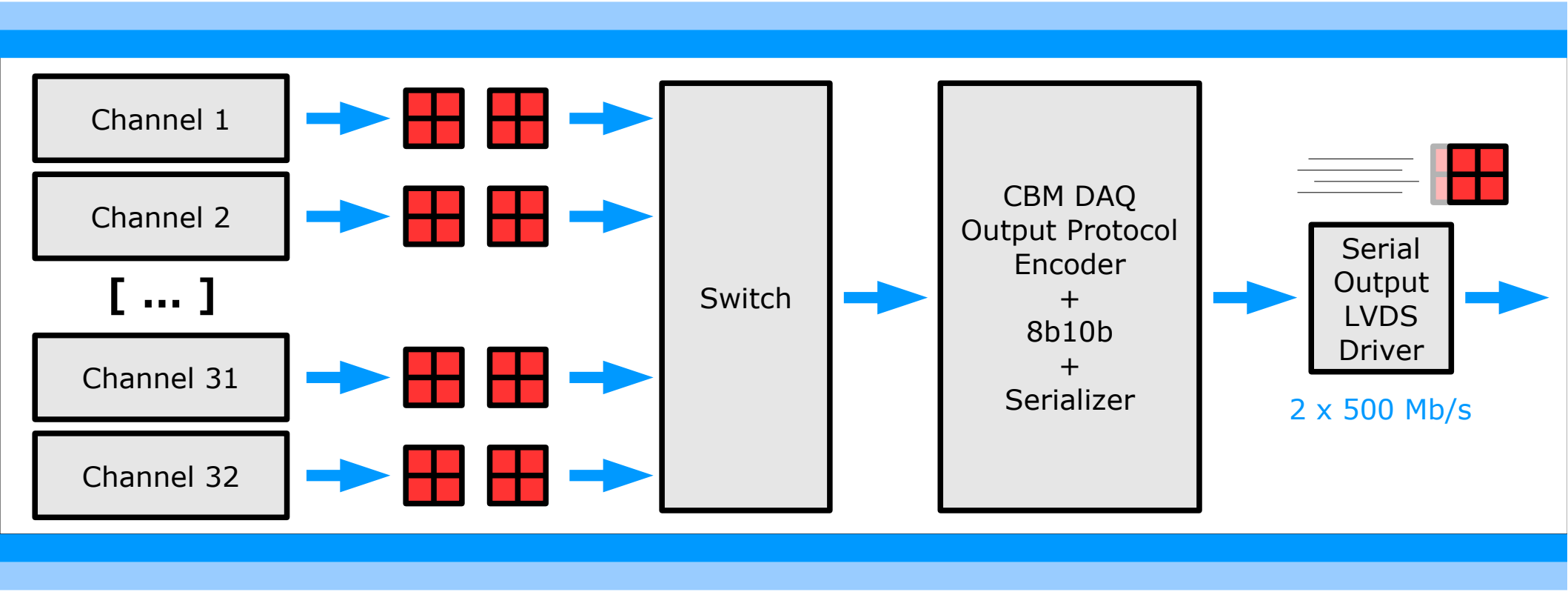
The processing chain of SPADIC 1.0 – Step 3

Step 3: Data processing + Data Gathering



The processing chain of SPADIC 1.0 – Step 4

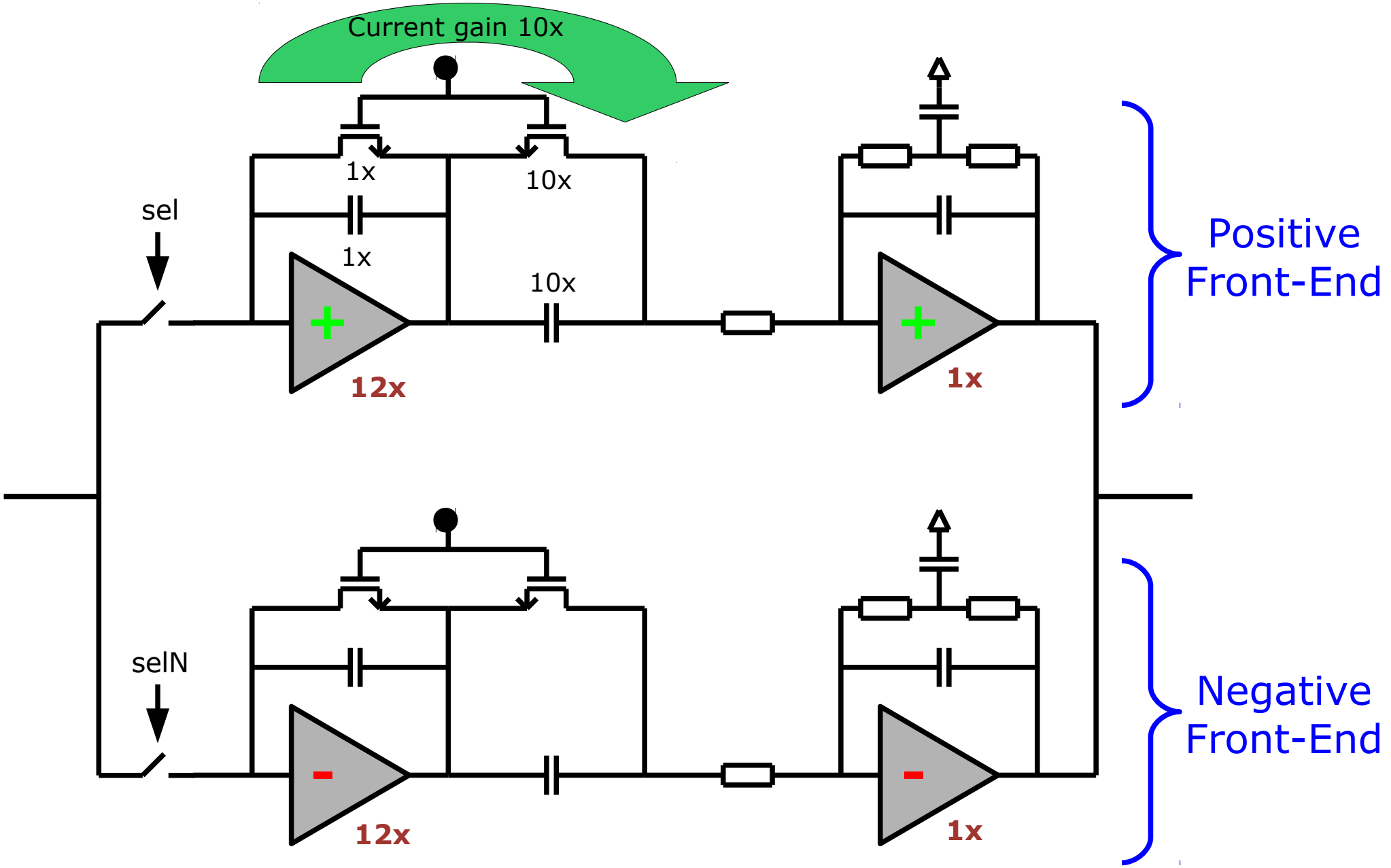
Step 4: Inter-Channel Network and Output Protocol



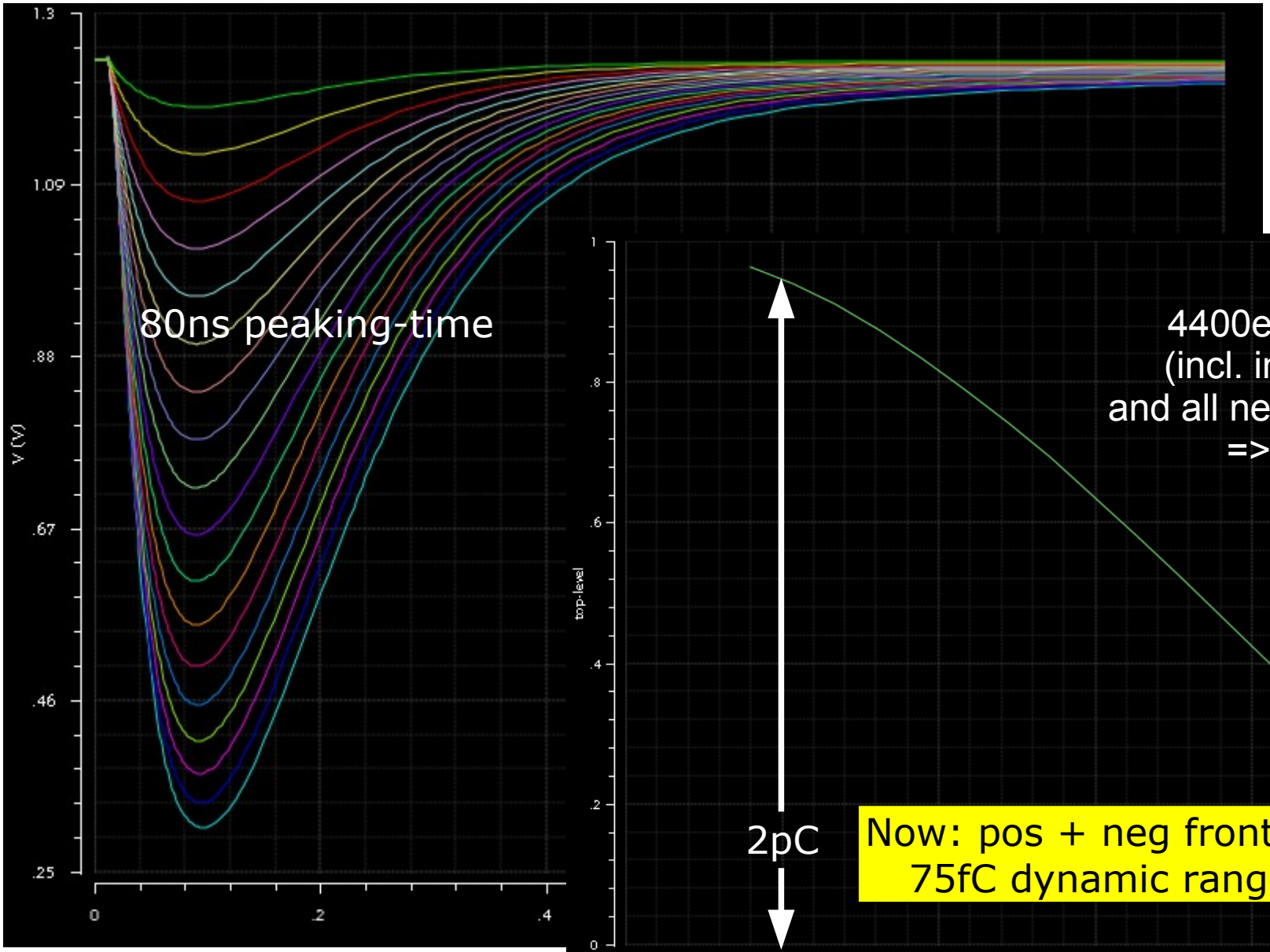
2. Selected Analog Details

(Since most of you already know a lot about the analog architecture)

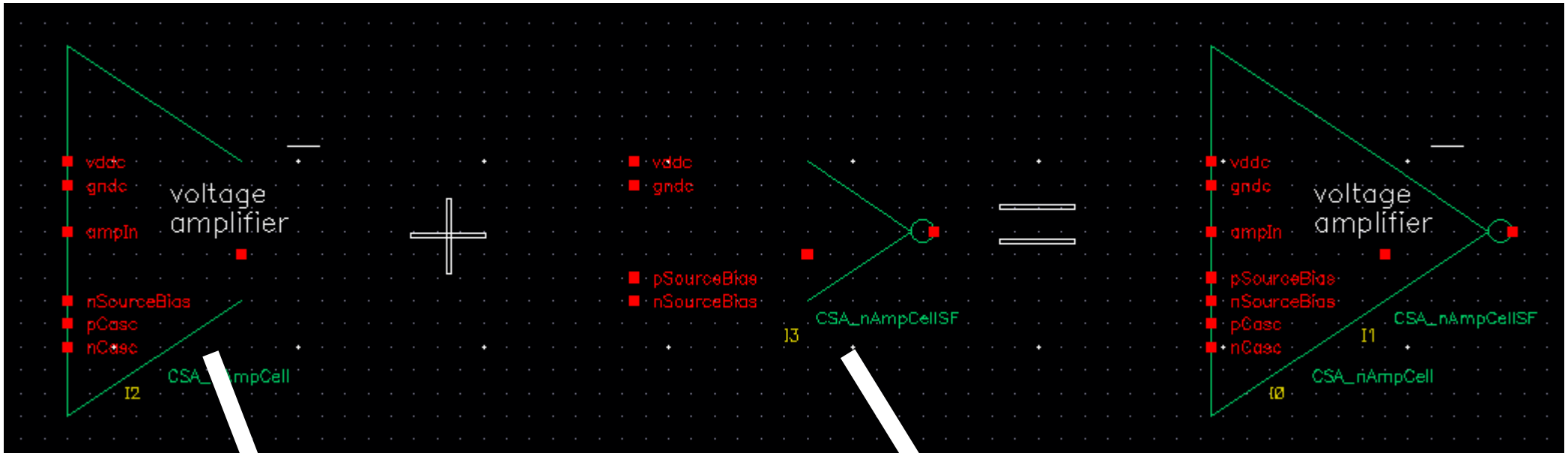
Two Front-End Amplifiers

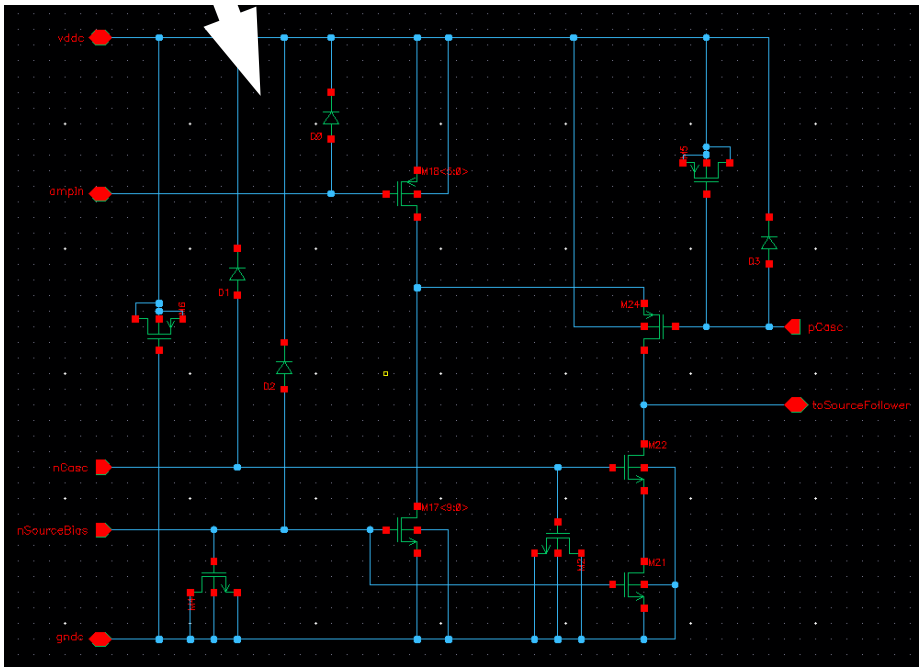


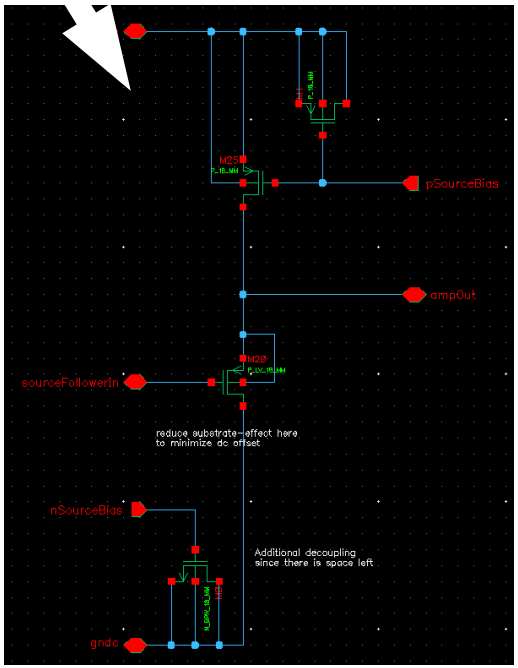
Negative Front-End (low vs. high gain = a long discussion)



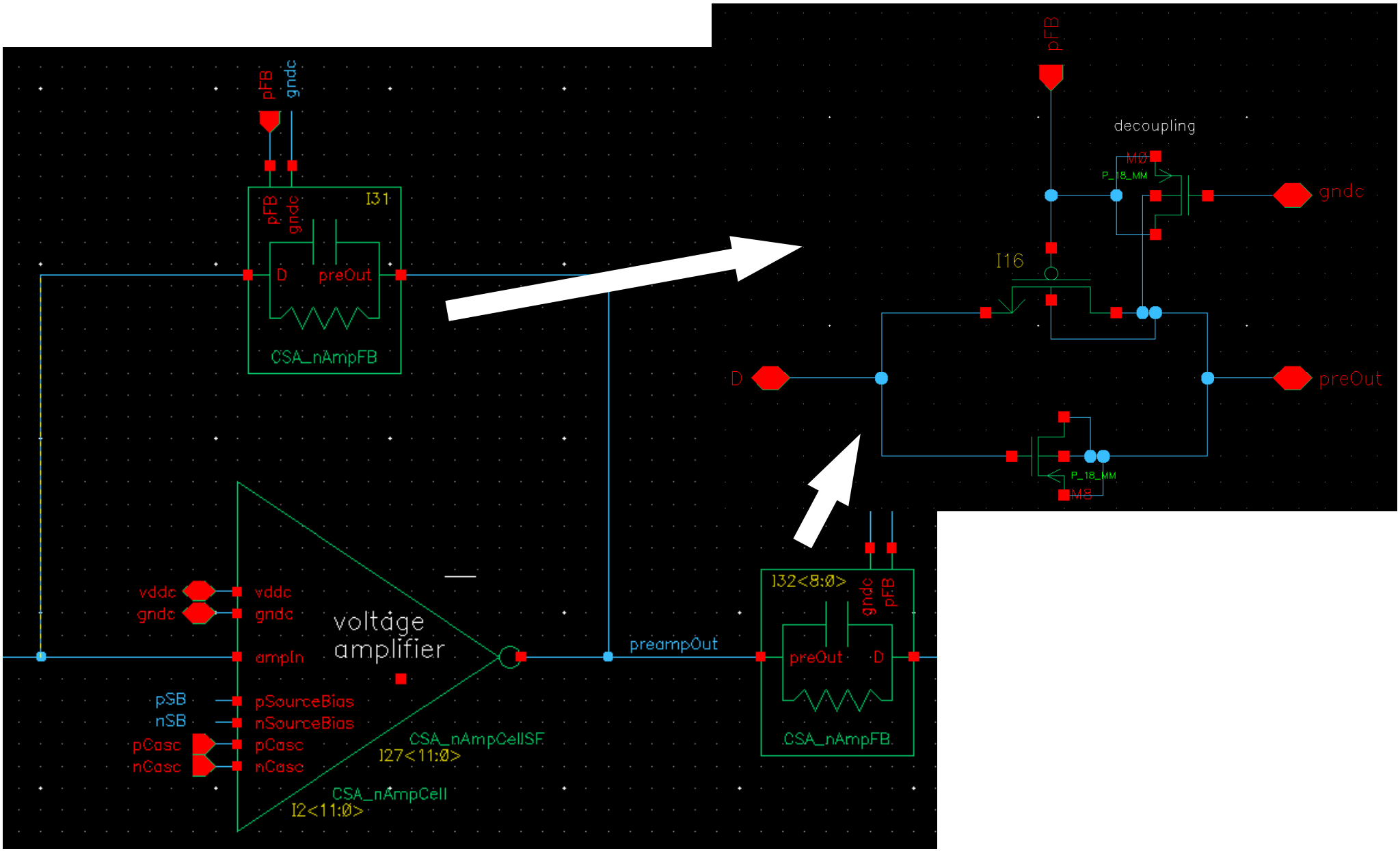
Modular Design – Example 1: Amplifier Cell



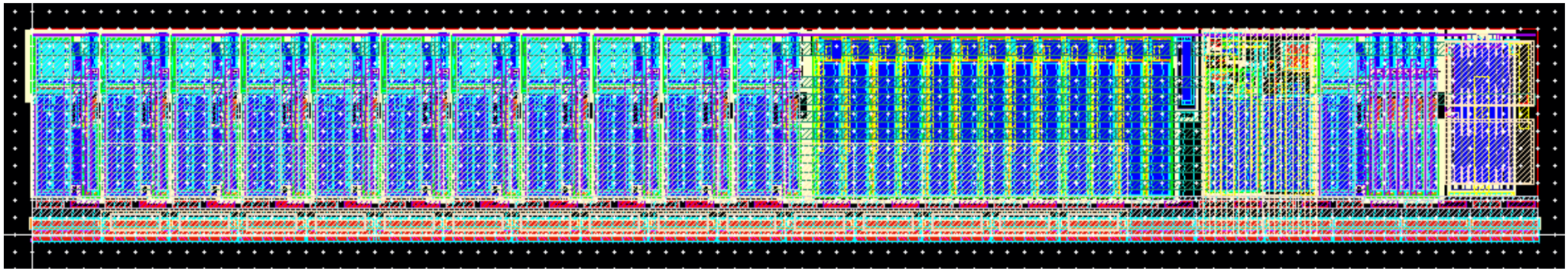




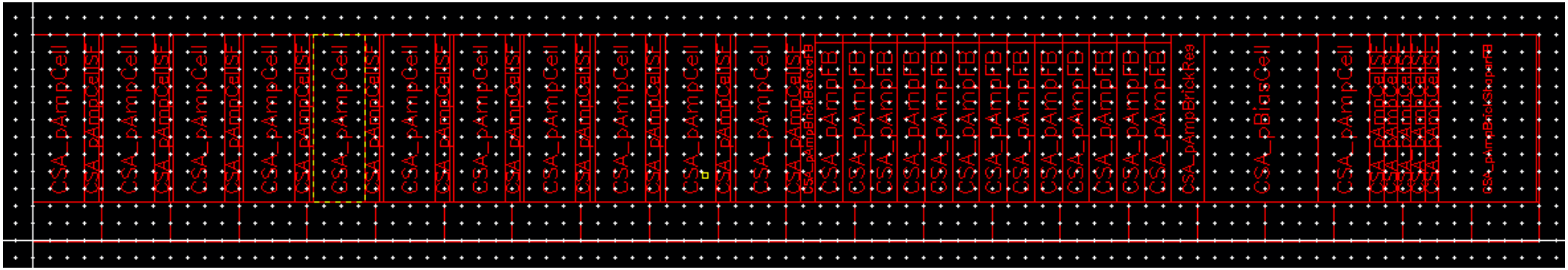
Modular Design – Example 2: Feedback



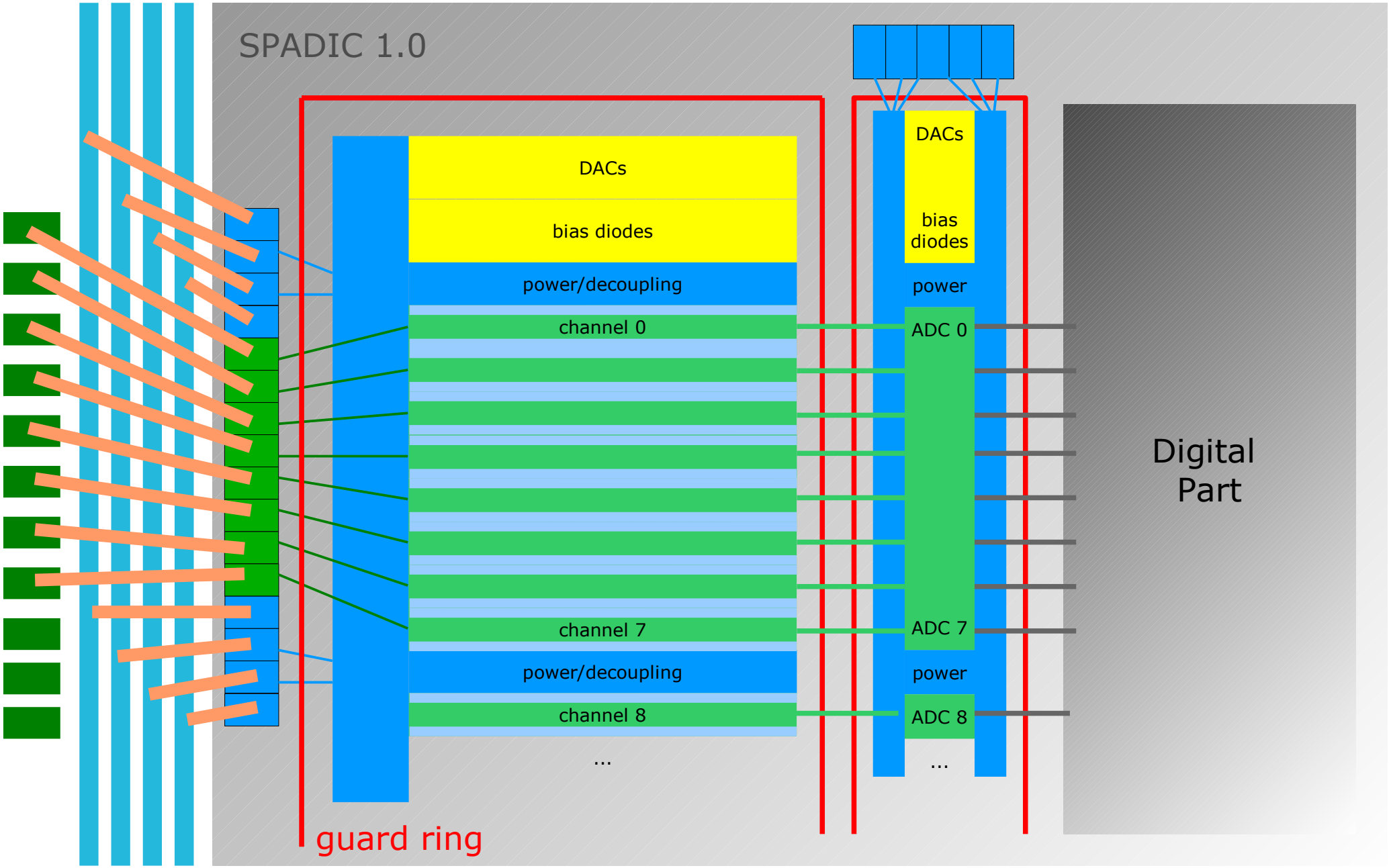
Modular Design – Example 3: Channel Bricks



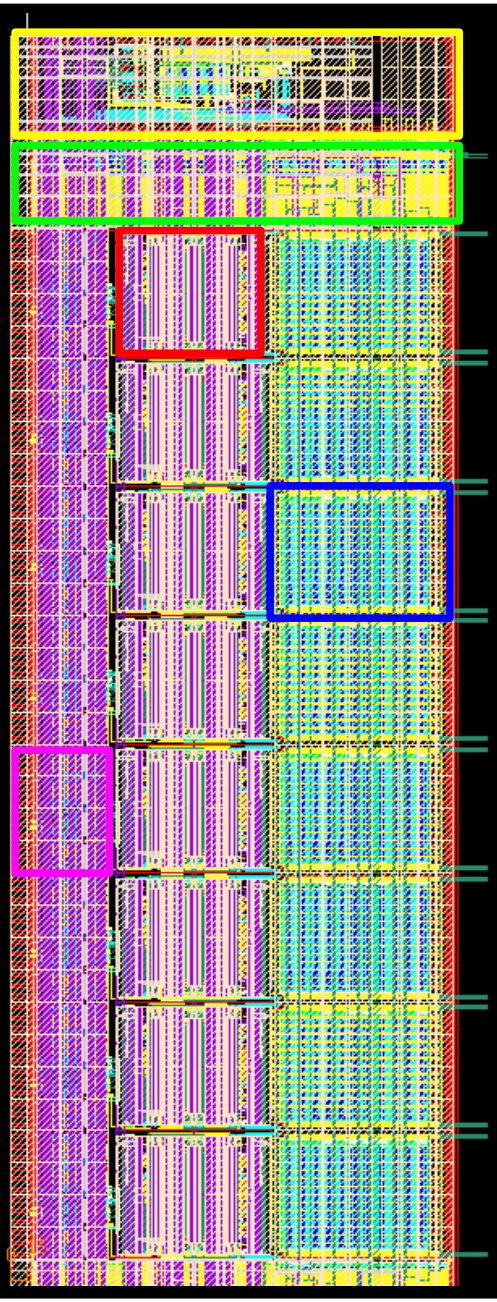
Modules wherever redundancy occurs
=> In all levels of hierarchy



CSA Power + Bias Scheme

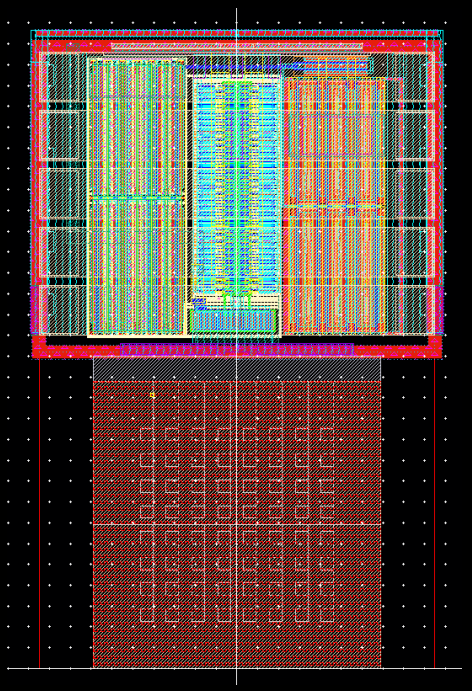


Analog Layout

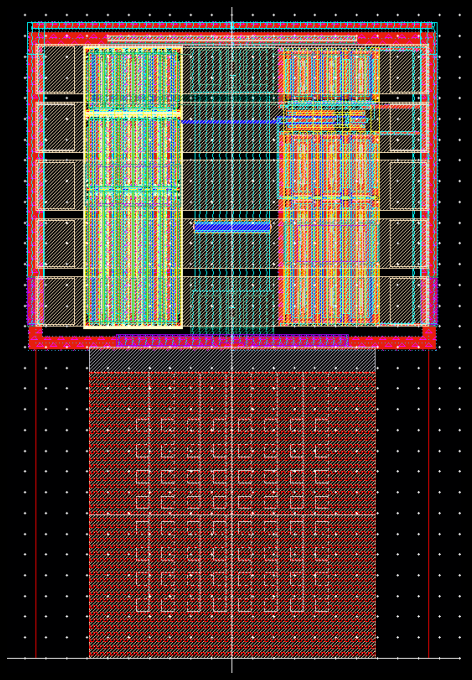


Pipeline ADC (32x)
Input Cell (32x)
Decoupling + Power Cell (32x)
Bias Diodes (1x)
Power channel + Decoupling (5x)

Pads



I2C



Protected

[...]

All SPADIC pads were copied from our own lib and then updated
Added some new pads: I2C, Analog Input with serial res, Pull-Up, …
New Pitch: 95um (old was 80um)

Some Numbers

2.2.1 Positive Front-End

Parameter	Design-/Sim-Value	Meas-Value	Comment
Peaking-time (0-100%)	86.7 ns @ 100 ke		
Input type	N-MOS		
Polarity	positive		
Dynamic Range	75 fC = 468.1 ke		
Power consumption	3.8 mW		
Size of layout	440 µm × 60 µm		
Number of amplifier cells	12		
Noise (@ 100 ke)	387 e + 11 e/pF		
Shaping-time	80 ns		
Order of shaper	2nd		
Order of complete CSA	1st (CR-RC)		
Pulse shape	f(t) = t/T exp(-t/T)		

2.2.2 Negative Front-End

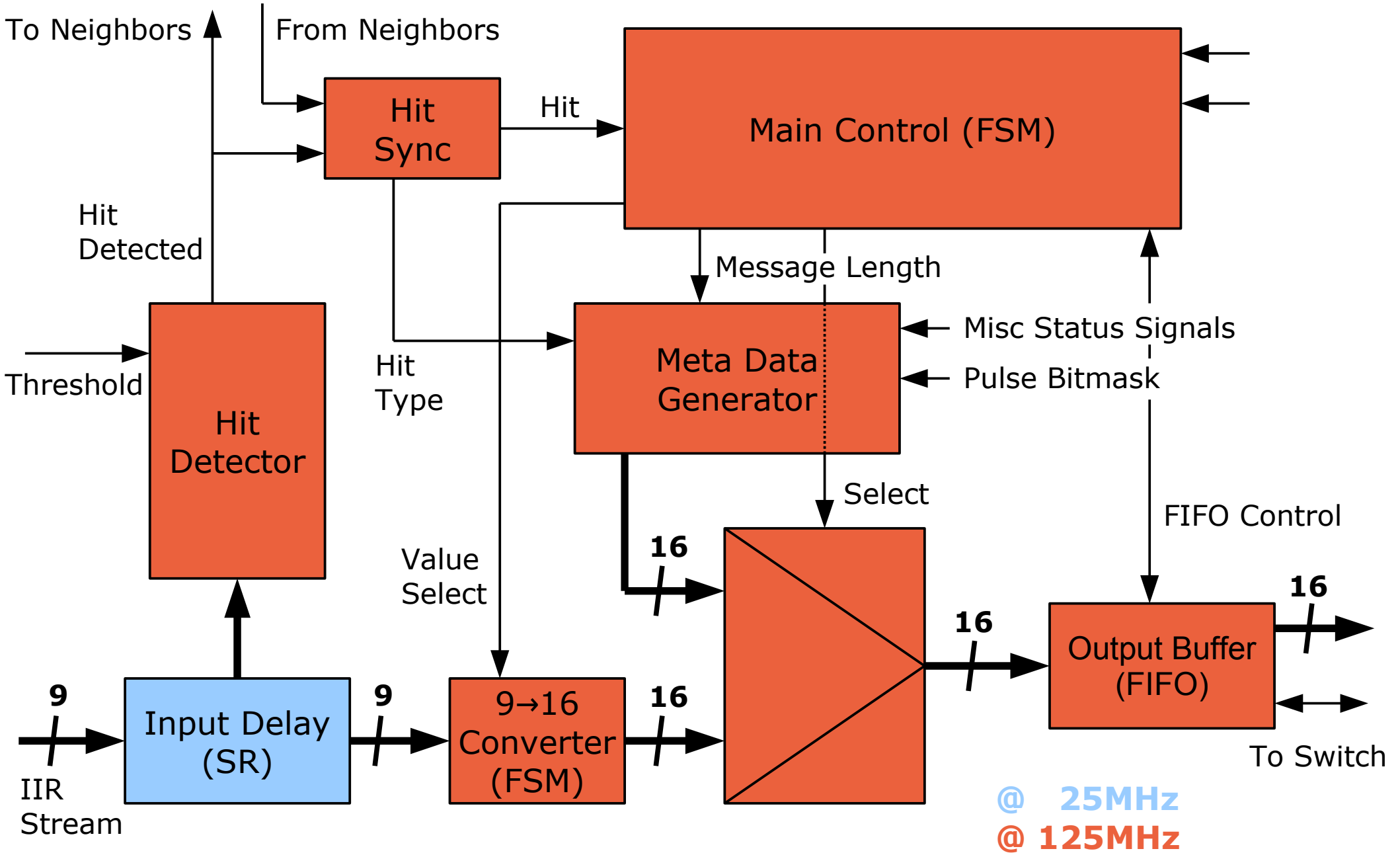
Parameter	Design-/Sim-Value	Meas-Value	Comment
Peaking-time (0-100%)	97 ns @ 100 ke		
Input type	P-MOS		
Polarity	negative		
Dynamic Range	75 fC = 486.1 ke		
Power consumption	10 mW		not optimized yet
Size of layout	440 µm × 60 µm		
Number of amplifier cells	13		
Noise (@ 100 ke)	439 e + 11 e/pF		
Shaping-time	80 ns		
Order of shaper	2nd		
Order of complete CSA	1st (CR-RC)		
Pulse shape	f(t) = - t/T exp(-t/T)		

2. Data Processing

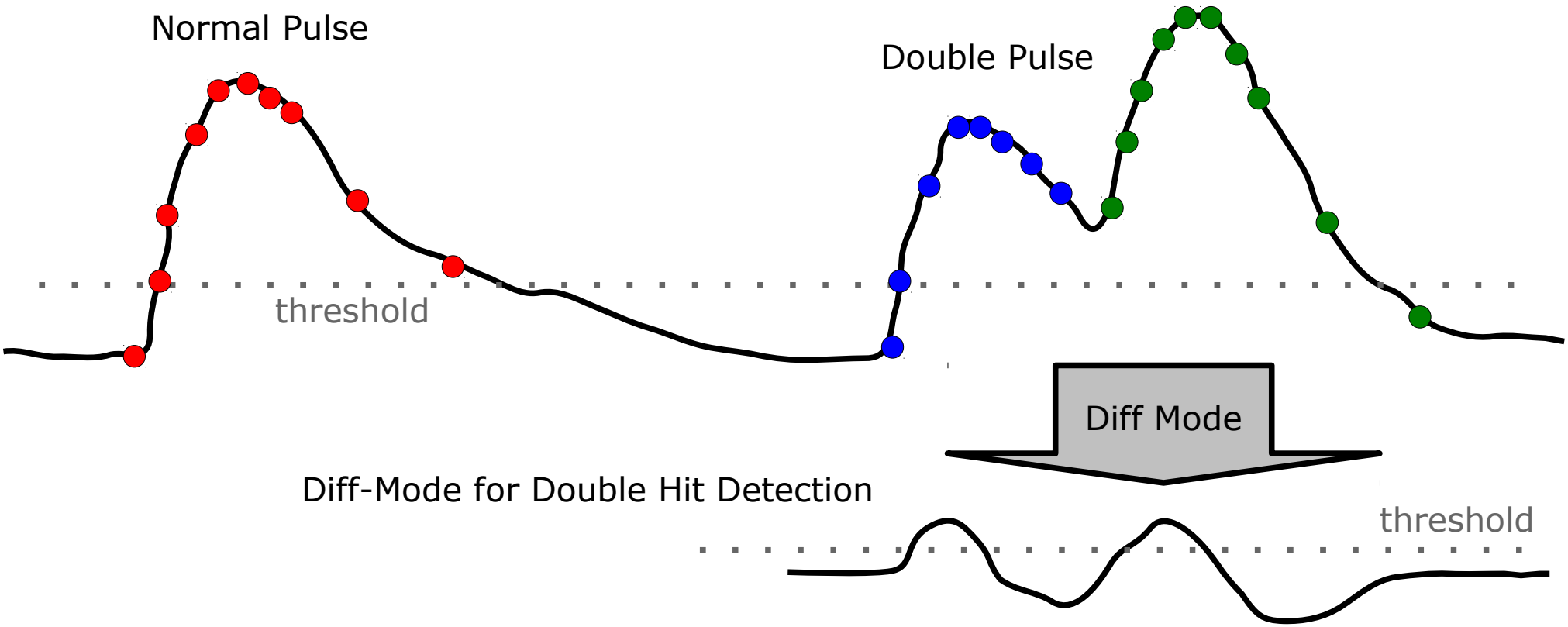
Now: Michael Krieger's Contribution

3. Data Extraction

Data Extraction Logic



Pulse Selection Mask + Double Hit Mechanism

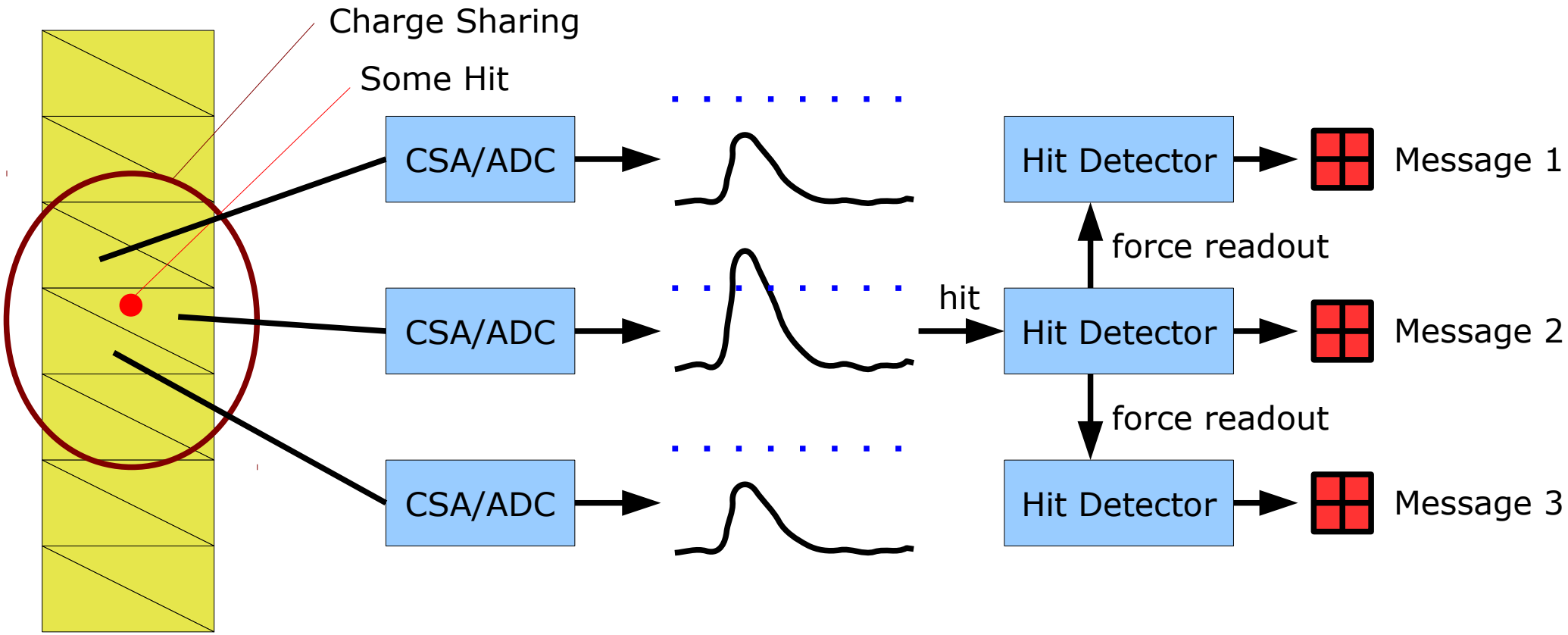


0000**11111111**00**1**0000**1**0000000000000000**11111111**00**11111111**00**1**0000**1**0000

Pulse Values Selection Mask

Restart of Selection Sequence

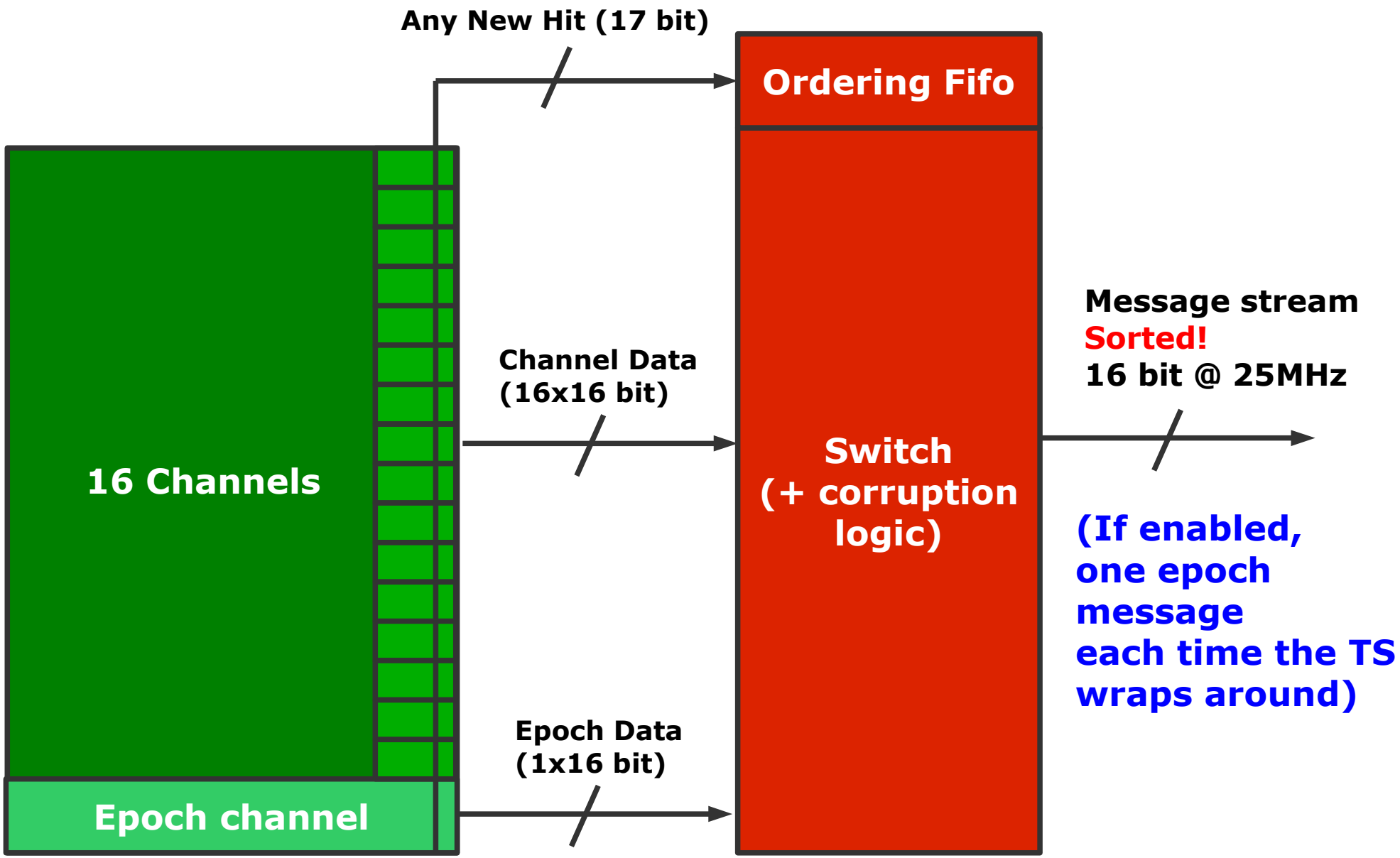
Neighbor Logic



=> Message 1: **Trigger from neighbor**, hit data, status, ID, …
Message 2: **Hit detected**, hit data, status, ID, …
Message 3: **Trigger from neighbor**, hit data, status, ID, …

Neighbor relationship programmable, force trigger also between chips

Channel Switch + Epoch Channel



Many Tricky Details

A lot of abort cases during massage building, e.g.:
- output buffer full
- ordering FIFO full
- multi hit detected

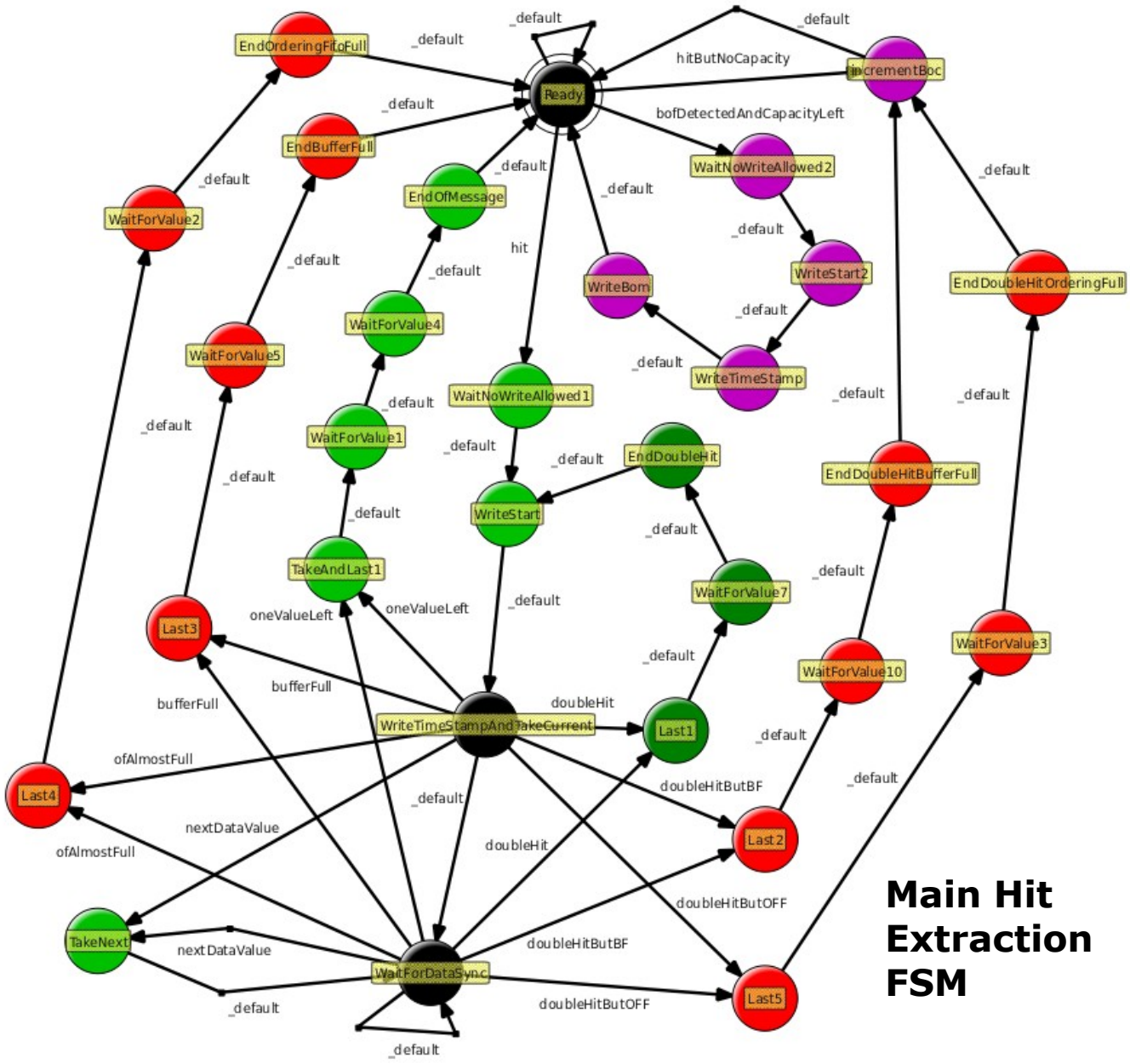
Counting mechanism if no further hits can be processed due to a full buffer
- hit but buffer full: inc hit counter
- counter > 0 && bufferNotFull: send message with counter value

SEU tolerance in switch
- bitflip in ordering fifo (SRAM)
- bitflip in output fifo (SRAM)

Data rate 25 MHz, internal Logic rate 125 MHz (1:5)
- No path in FSMs longer than 5

9 bit data width but 16 bit message word width
- Very tricky converter 9 → 16 bit

[...]



Main Hit Extraction FSM

Message Format

4.3 Example Messages

normal hit message (n-word message)		
1000 ssss ssss iiii	start of message	
1001 tttt tttt tttt	time-stamp	
1010 dddd dddd dddd	start of raw data	
0ddd dddd dddd dddd	continued raw data	
[...]	continued raw data	
0ddd dddd dddd dddd	continued raw data	
1011 pppp pphh -sss	end of message	
buffer overflow message (3-word message)		
1000 ssss ssss iiii	start of message	
1001 tttt tttt tttt	time-stamp	could be removed later in stream
1100 -- bbbb bbbb	buffer overflow counter	end of message
epoch marker message (2-word message)		
1000 ssss ssss 0000	start, channel-id fixed to 0 (necessary for arbitration)	can be removed later in stream
1101 eeee eeee eeee	epoch marker	end of message
extracted hit message (n-word message)		
1000 ssss ssss iiii	start of message	
1001 tttt tttt tttt	time-stamp	
1110 dddd dddd dddd	start of extracted data	
0ddd dddd dddd dddd	continued extracted data	
1011 pppp pphh -sss	end of message	
empty word info message (1 word only)		
1011 pppp pphh -sss	end of message	

Message word: 4 bit preamble + 12 bit payload
OR preamble "0" + 15 bit payload (continuation word)

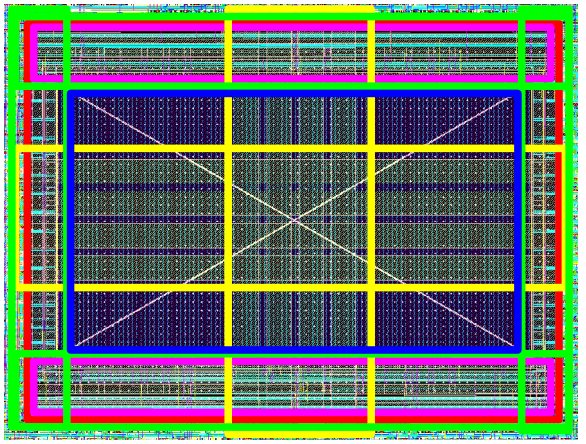
Message: Block of n message words (e.g. normal hit message, epoch marker message, …), may consist of only 1 message word

=> Very flexible, in case of SEU message stream can always be re-synchronized

Long story: Using the Faraday SRAMS

Very few documentation available

→ More or less only the manual for the RAM generator software which we can not access directly...



Many complicated constraints where necessary to tell the tools not to short wires, to connect power, ...

→ Manually added blockage layers (a lot of trial and error)
→ Cut out standard cells (find the correct command sequence)
→ Manually tell encounter to connect the power
→ Choose proper size of guard rings

44 SRAM blocks made the floorplanning to a big challenge

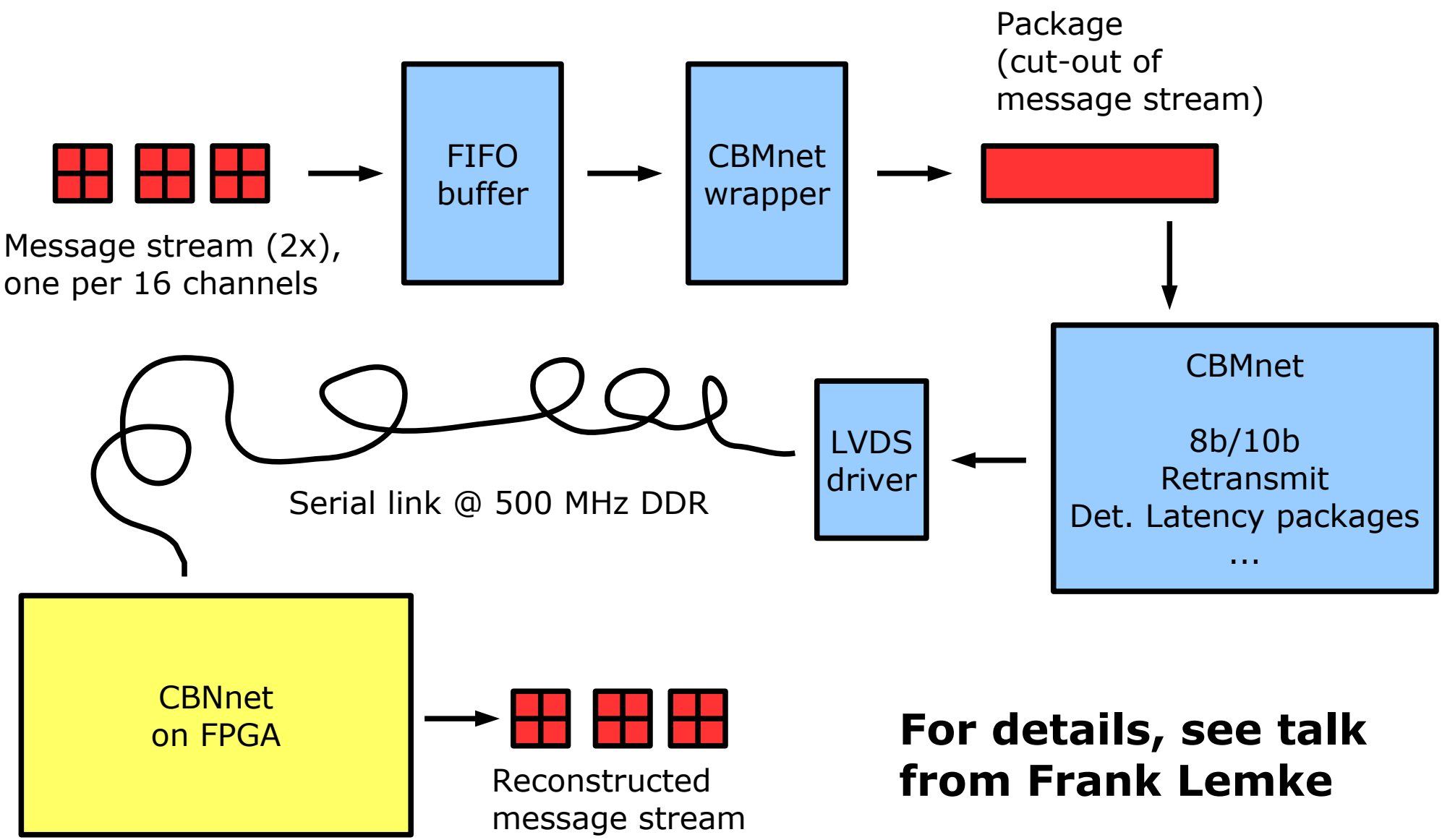
→ 5 completely different layout schemes were necessary
→ Whether the router succeeds or not seems to be randomly correlated to the floorplan

LVS was a mess

→ LEF input very buggy, needed to write auto-repair SKILL scripts
→ At the end, LVS only with digital part as black box

4. Transfer Mechanism

CBMnet from a SPADIC's point of view



For details, see talk from Frank Lemke

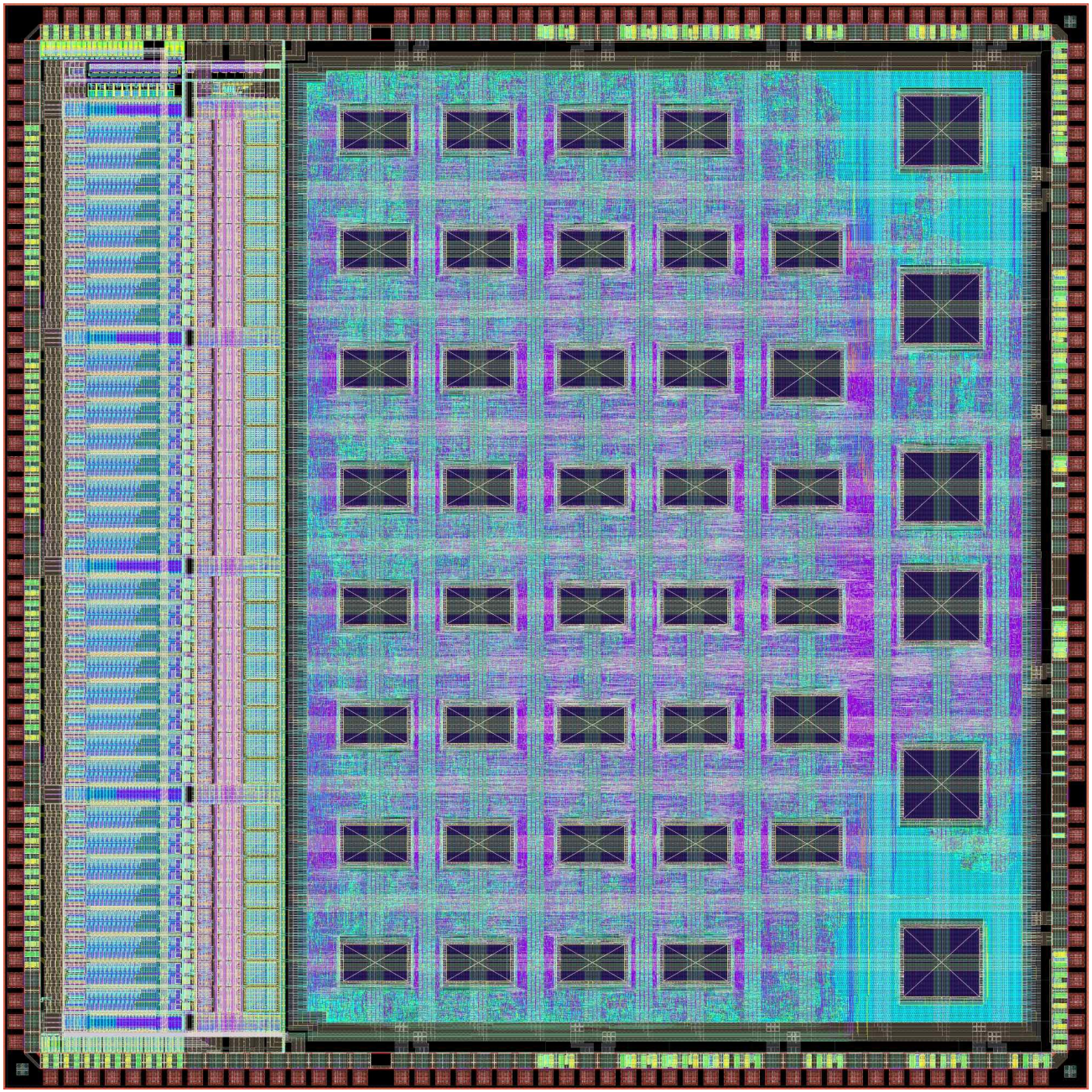
Synchronization

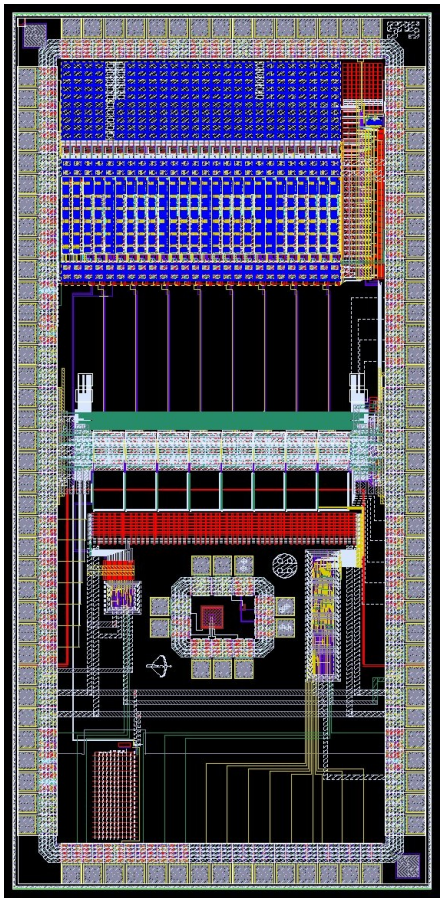
CBMnet feature: DLM (deterministic latency messages)

From SPADIC's point of view:
DLM0: Loop back (used to measure delay through CBMnet)
DLM1: Sync signal (always @ TS wrap-around, set new TS + opt: epoch)
DLM2: With next DLM1 set TS
DLM8: Start readout
DLM9: Stop readout
[...]

4. Summary

Top-Level Layout





SPADIC 0.3
(1.5 x 3.2 mm²)

SPADIC 1.0 (5 x 5 mm²)

The Documentation: e.g. Some Statistics

2 SPADIC 1.0 Documentation

Just google "SPADIC" :-)

2.1 General

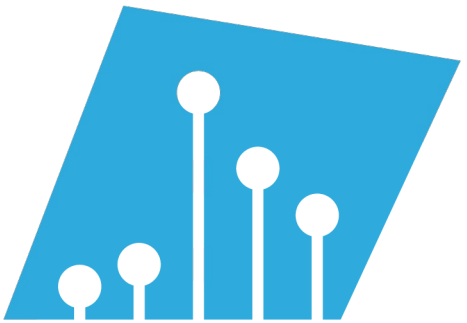
Parameter	Design-Value	Meas-Value	Comment
Technology	UMC 180 nm		
Number of metal layers	6		
Number of channels	32		
Channel pitch	120 µm		
Pad pitch	95 µm		
Chip size	4.96 mm × 4.96 mm		3 × 3 Mini@sics
Total power consumption			
	Analog Part	Digital Part	Total
Number of transistors	226 k	2.09 M	2.32 M
Number of nets	58.1 k	1.02 M	1.08 M
Number of pads			191
Number of sram blocks		44	44
Total sram memory		4.4 kByte	4.4 kByte
Number of DACs	48		48
Number of config registers	576	1270	1846
Total wire length		14.39 m	
Area	1.12 mm × 4.65 mm	3.46 mm × 4.54 mm	4.96 mm × 4.96 mm

Summary / Next Steps

Next Steps
- Build (low pickup noise!) PCB
- Test and provide CBMnet on FPGA (ongoing, Brüning et al)
- Wait for SPADIC 1.0 to return from factory

What SPADIC 1.0 is (if it works)
- An oscilliscope-like charge pulse readout ASIC
- Complete multi-channel mixed-mode readout *system*
- Flexible, adjustable and programmable
- Simulated

What SPADIC 1.0 is **not**
- An amplification wonder (noise, power, …)
- A digitization wonder (8 bit only)
- A minimalist (full pulse readout, high bandwidth, ...)
- An answer to all possible problems (SEE, fixed shaping time,…)
- Tested



SPADIC

Self triggered Pulse Amplification and Digitization asIC

http://www.spadic.uni-hd.de