Solving the idle power problem of a multi-core software defined radio

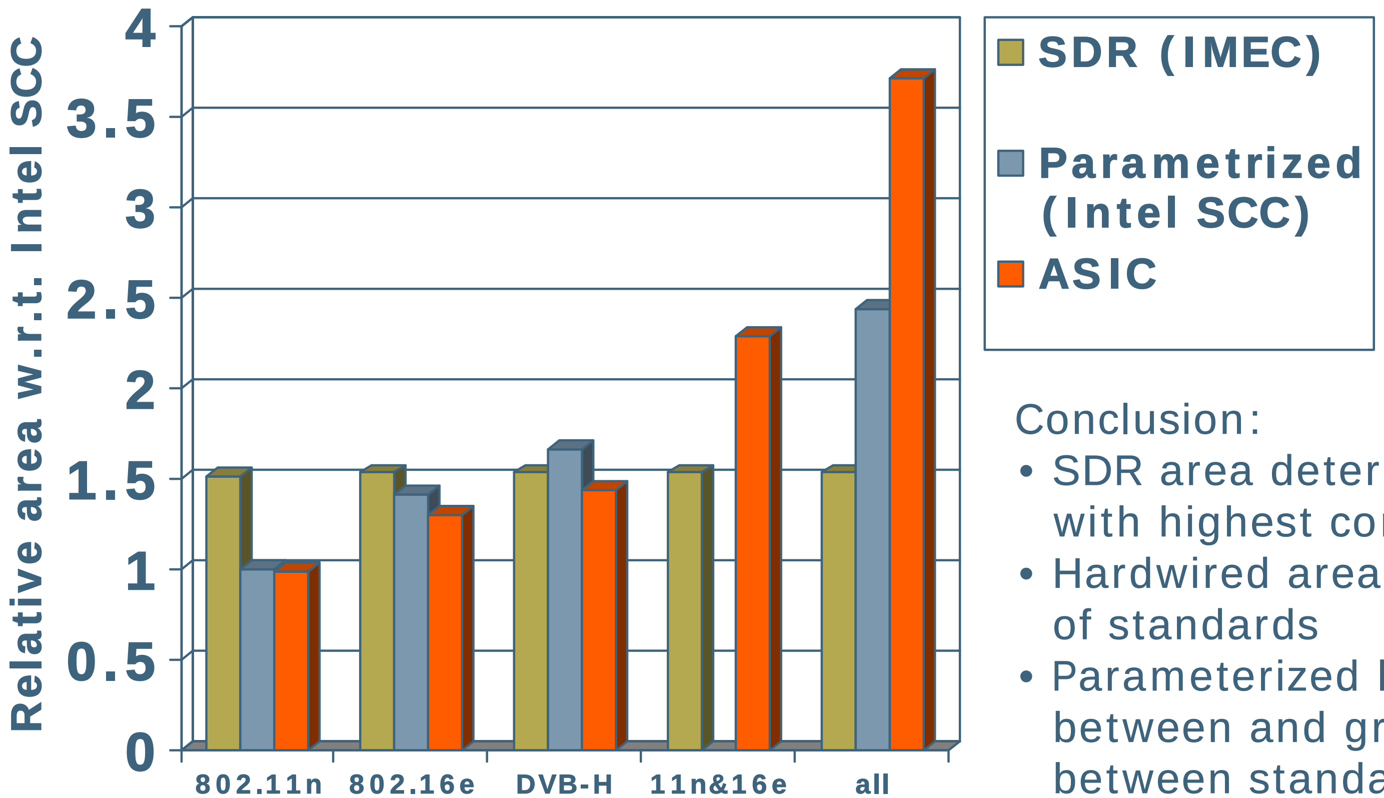
Rudy Lauwereins

VP Nomadic Embedded Systems, IMEC
Prof. Katholieke Universiteit Leuven

FLAI (Flexible Air Interface) aims for energy efficient multi-mode SDR terminals

- Cost of implementing multiple standards in one device asks for SDR (Software Defined Radio)
- Energy efficient SDR requires heterogeneous MPSoC with Targeted Flexibility

Comparing SDR with (parameterized) hardware: comparing IMEC SDR to Intel SCC and to ASIC



Conclusion:

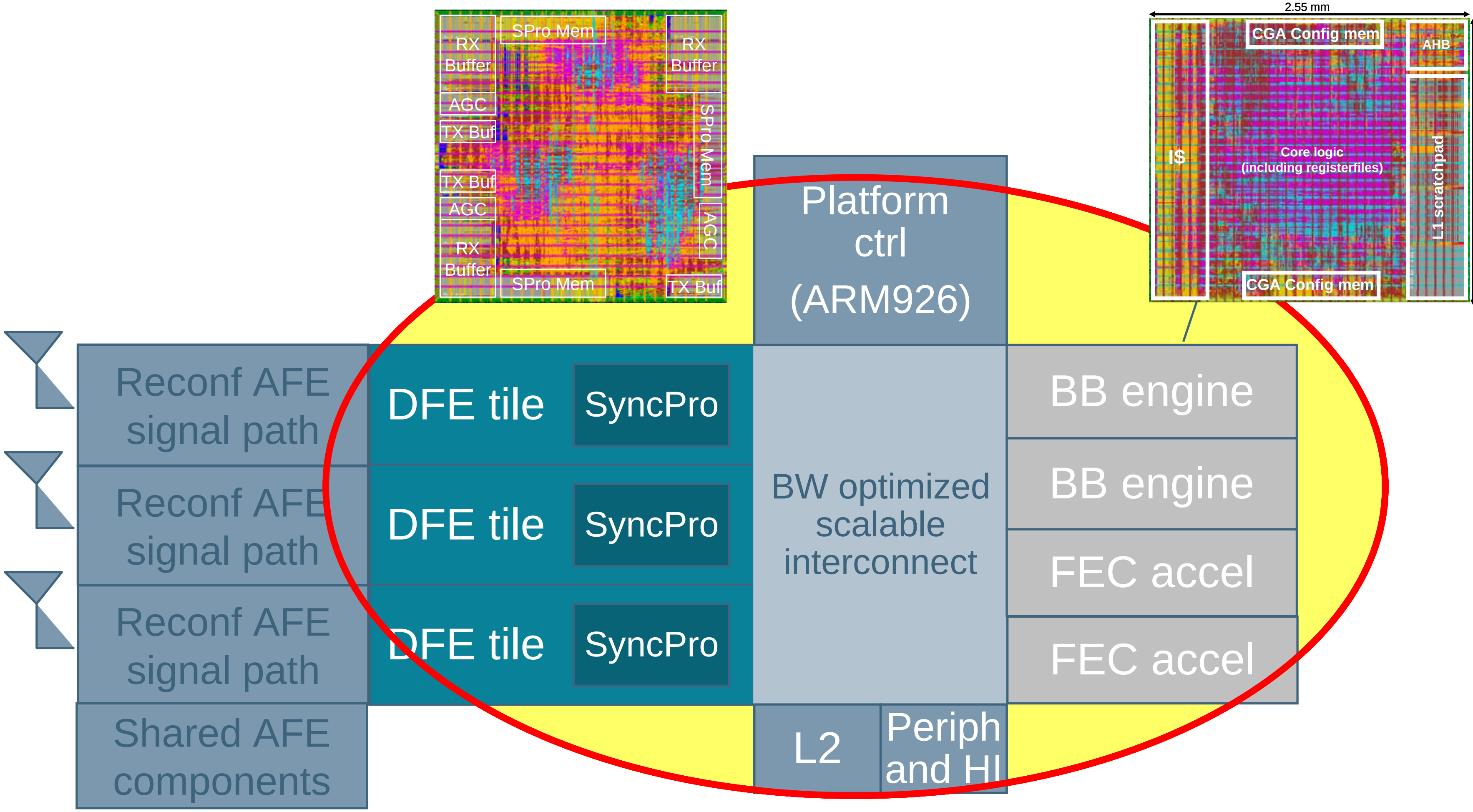
- SDR area determined by standard with highest computational demand
- Hardwired area determined by sum of standards
- Parameterized hardware area sits in between and grows with differences between standards

Additional advantages of SDR

- SDR only way to implement future flexible standards
- SDR does not need re-design between pre-standard, draft standard and “final” standard
- SDR allows for cheap bug fixes, regional adaptation

IMEC's SDR baseband platform:

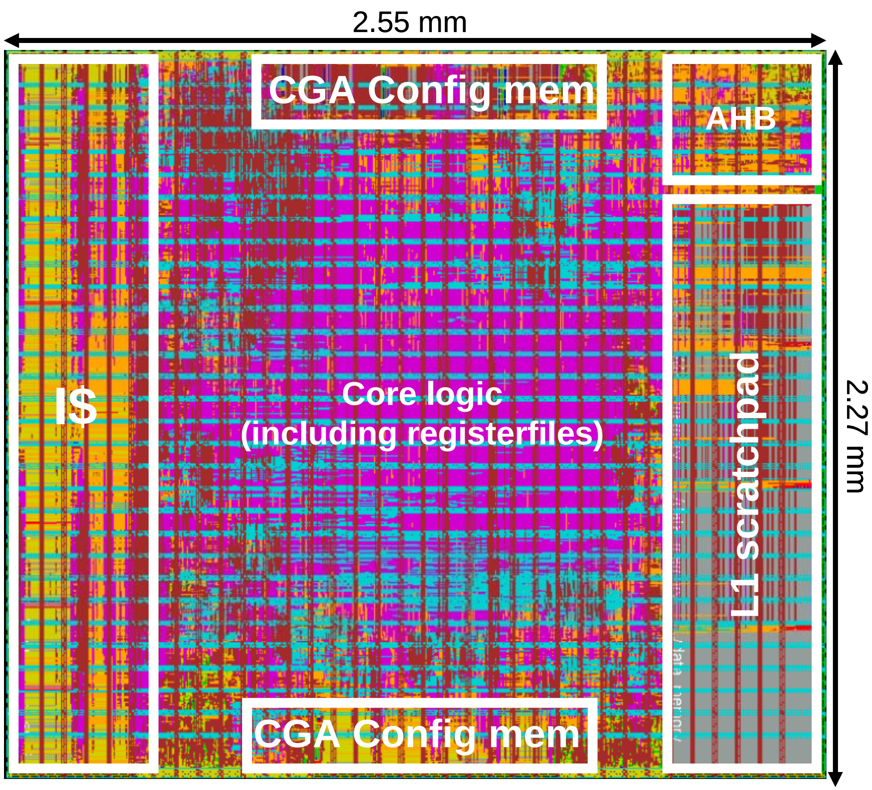
Heterogeneous MPSoC enabling reactive radio



Instantiated for

3GPP-LTE, 802.16e and 802.11n

ADRES was finalized with VST libraries



- 32KB instruction cache
- 128KB Instruction mem
- 128-entries config mem
- 128KB data scratchpad

- TSMC 90G
- Dual VT and substrate biasing for leakage reduction in sleep mode

- Clock rate 400MHz WCC
- 25 GOPS peak

- Total Area: 6 sqmm
- Power estimation (gate level, after PR)
  - Active TC 1D-VLIW 75mW
  - Active TC 2D-VLIW 300mW
  - Leakage @ T=65C 25mW
  - Leakage in standby ≤10mW

Where does the power go?
Where do you need flexibility?

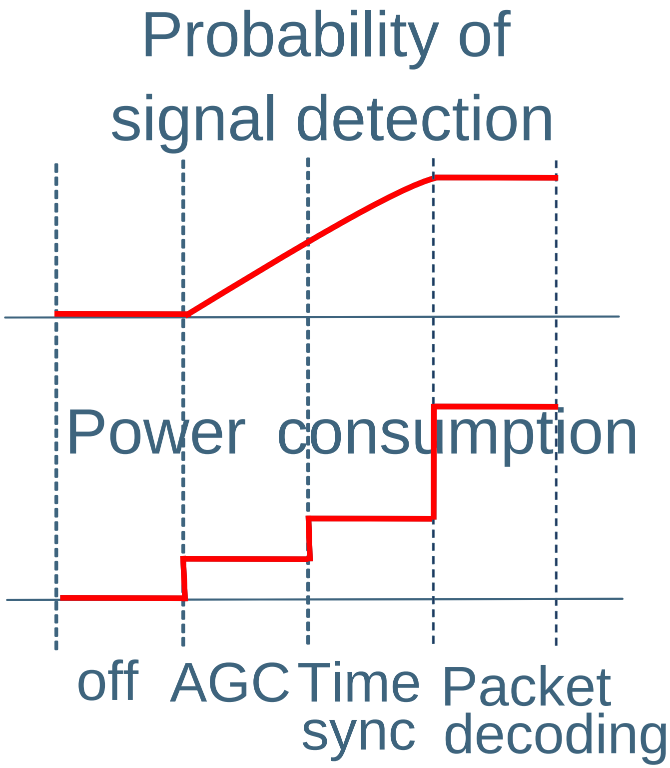
Typical Wireless LAN:		Typical Cellular:	
Transmit	5%	Transmit	.5%
Receive	5%	Receive	.5%
Idle/Listen	90%	Idle/Listen	99%

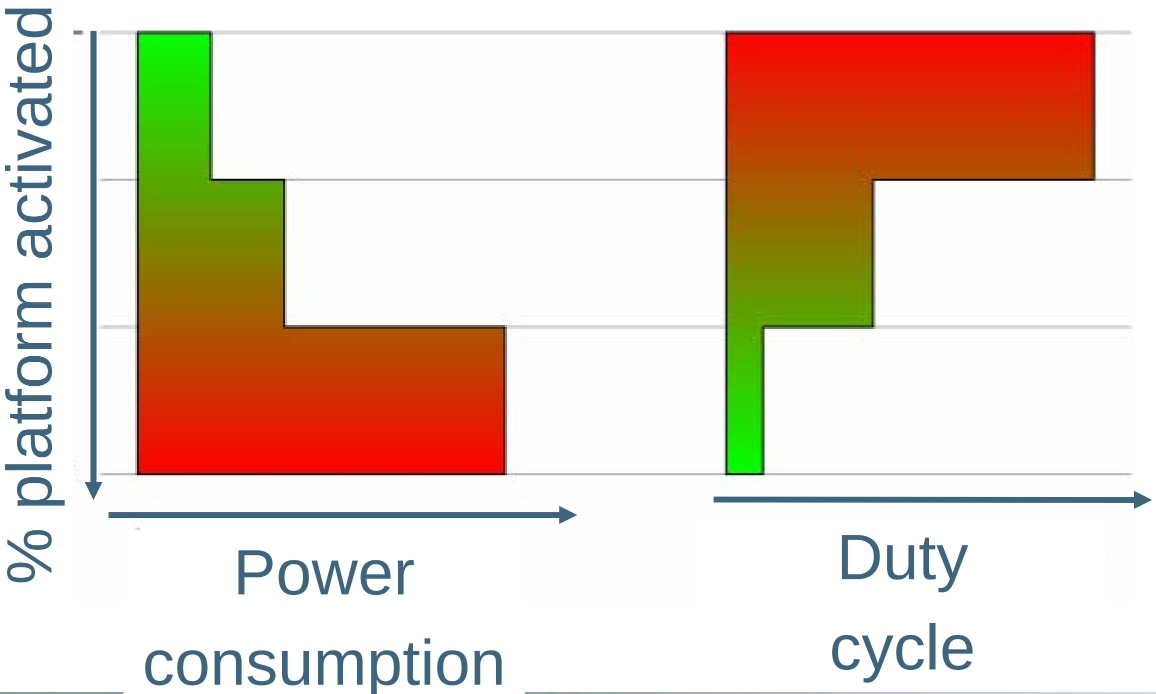
Approach: 'Just enough' tuned flexibility

Divide and conquer

Hierarchical activation: guarantee low power reactive radio

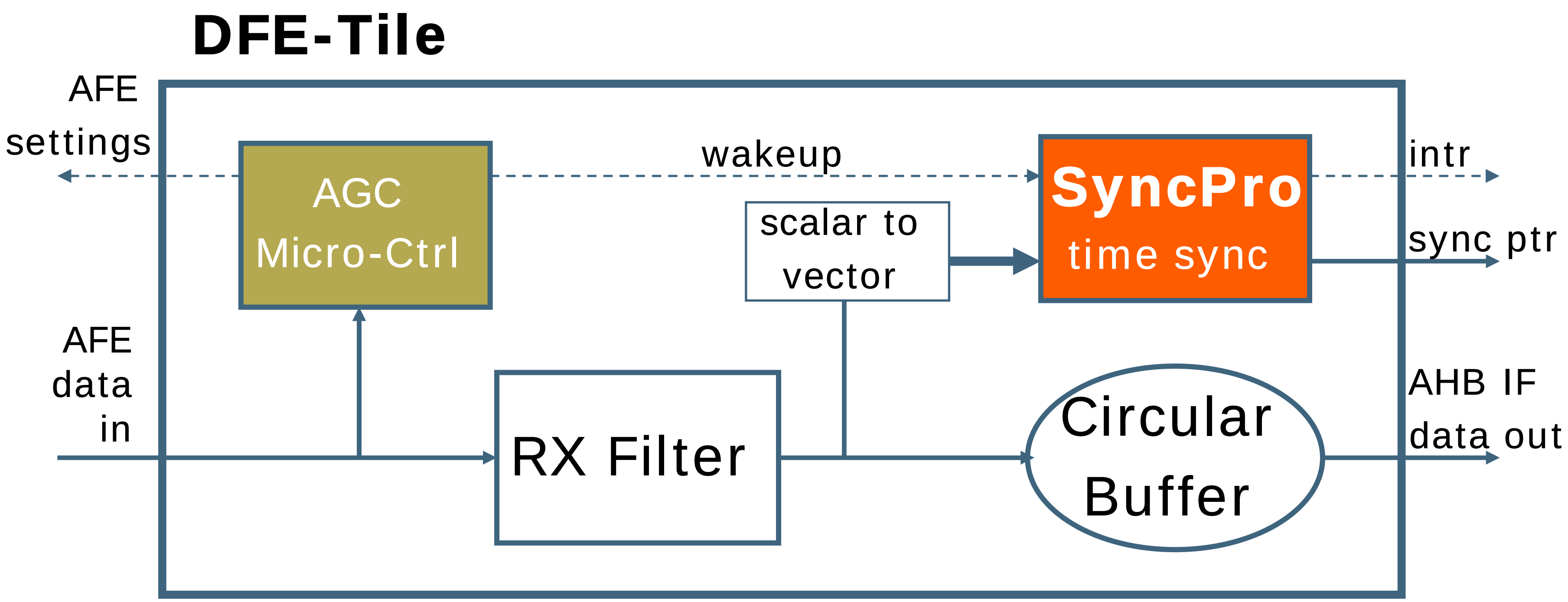
Gradually enable more power-consuming parts as the chance of a valid signal reception increases





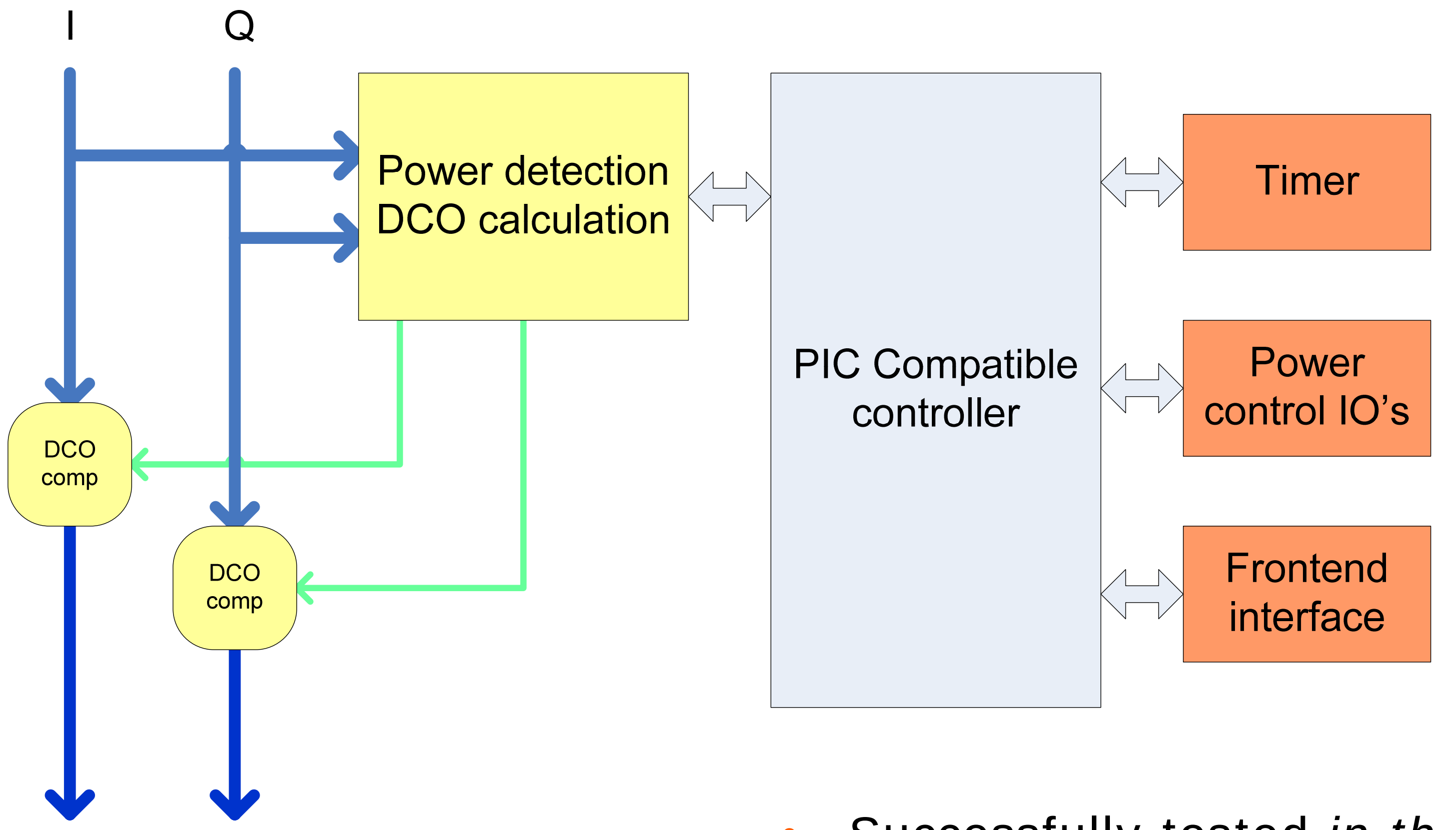
Patent filed

The Digital Front End consists of the Automatic Gain Control, Receive filter, Synchronization Processor and Buffering



Exploit: Signal Detection and Time Synchronization have higher duty cycle than modem functions.

A zoom on the AGC



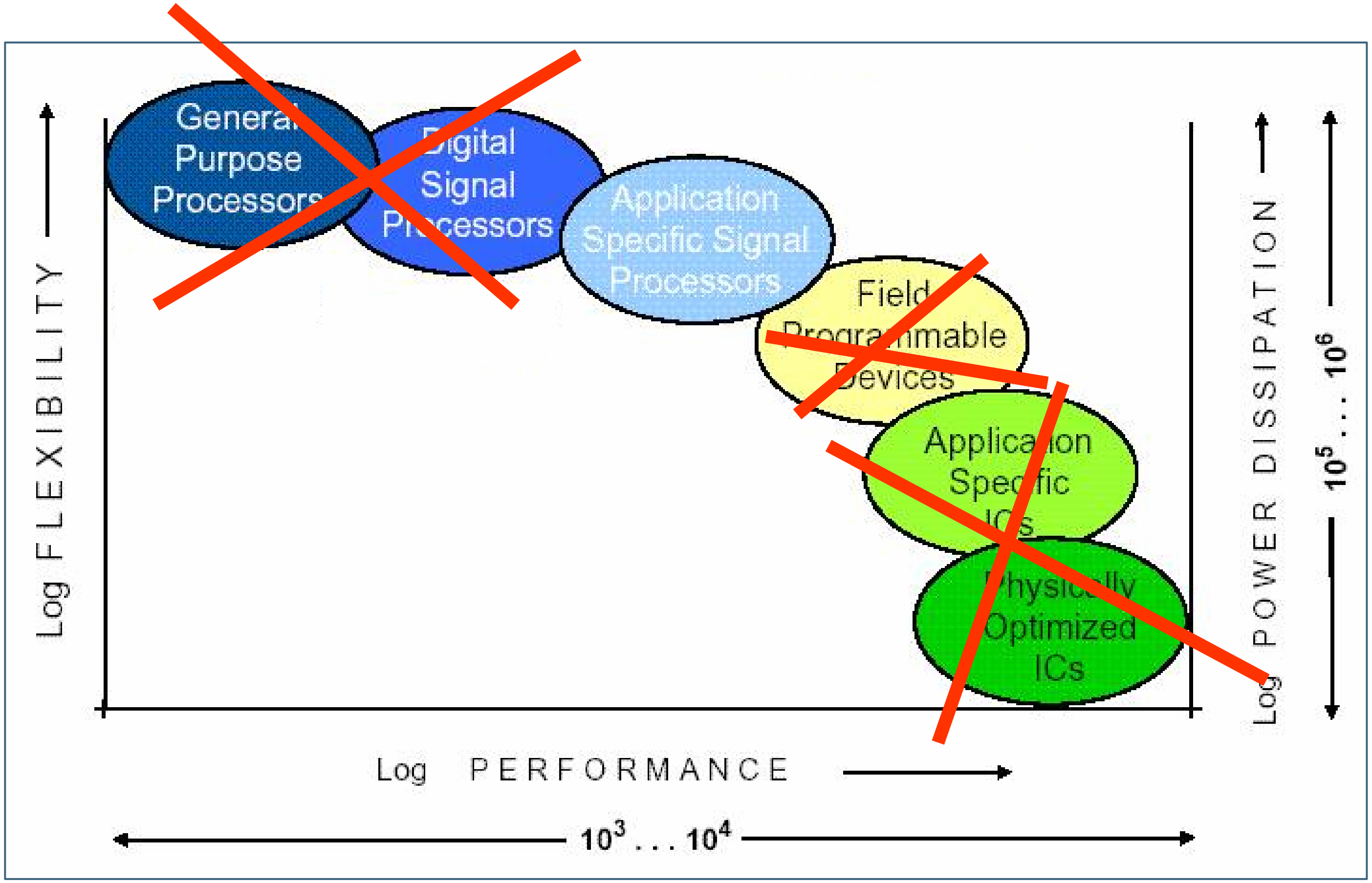
- Successfully tested *in the air* for WLAN detection
- Estimated power: 1.1mW

SyncPro – Design Goal/Constraints

- IEEE802.11a/g/n, IEEE802.16e time synchronization with up to 20MHz input rate
- Provision for implementation of other standards (e.g. 3GPP-LTE)
- Clock rate should be derived from 200MHz
- 90nm CMOS target tech
- power budget 20mW (25C, 1V)

What is the right architectural approach ??

ASIP will deliver the best energy/efficiency tradeoff



Architecture Definition for SyncPro ASIP

Instruction-Set Selection

↓

Parallel Processing

↓

Clustering and Interconnect

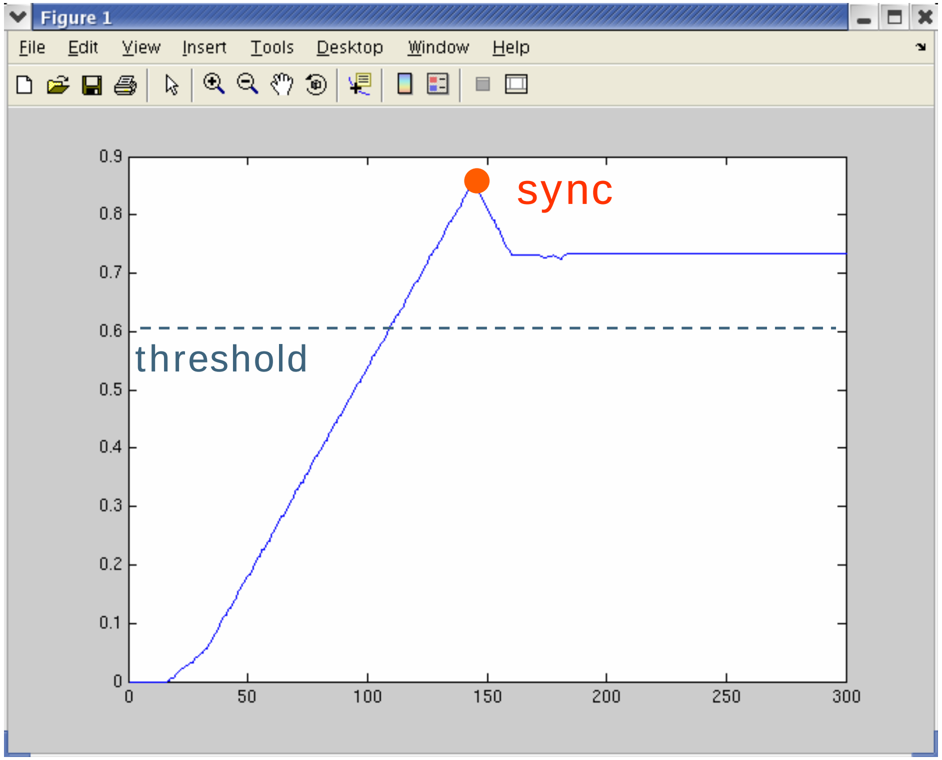
↓

Memories, Pipelining

↓

Implementation

ASIP design starts with analysis of algorithms



802.11a synchronization peak

OFDM(A) Time synchronization:

1. correlate over input signal to expose periodic structure of packet preamble
2. find peak in correlation results, above certain threshold

Code consists of data and control intensive parts !!

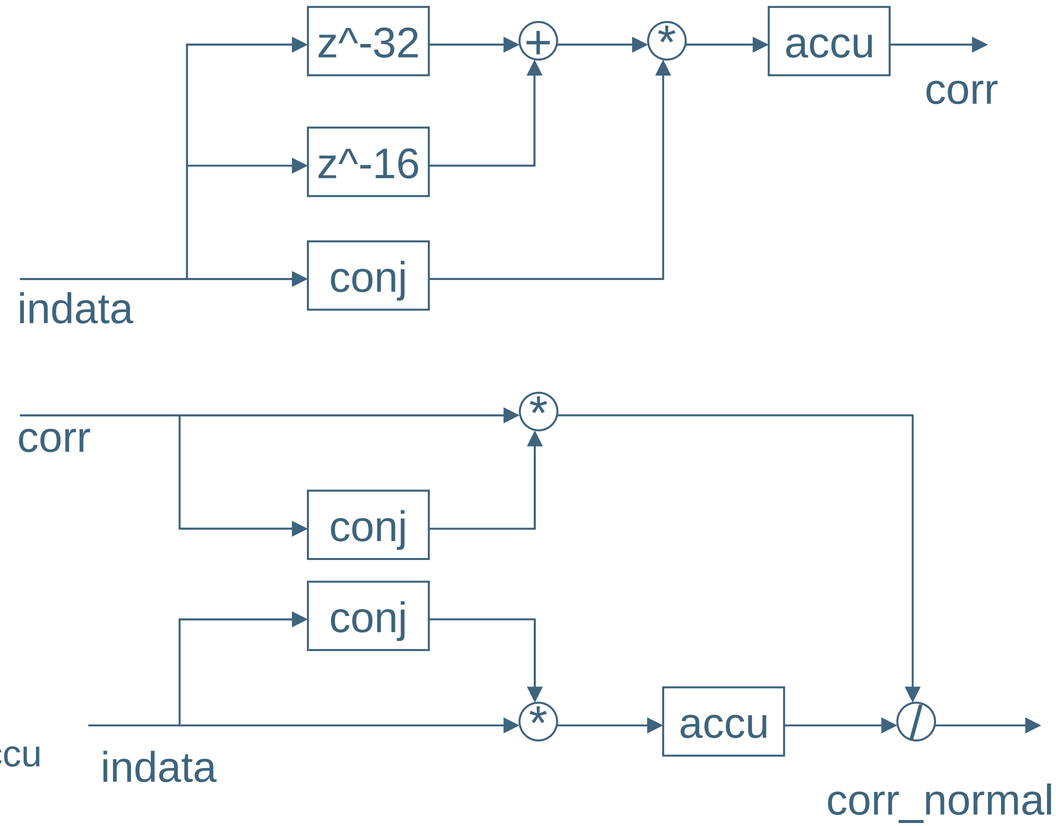
Autocorrelation contains high Data Level Parallelism (DLP)

```
/* LOOP 1 – CORRELATE (data) */

for i = START:END
   sample = indata[i]
   sample16 = indata[i-16]
   sample32 = indata[i-32]
   sum = sample16 + sample32
   prod = sum * conj(sample)
   corr[i] = accumulate(prod)
end

/* LOOP 2 – NORMALIZE (data) */

for i = START:END
   sample = indata[i]
   power = sample * conj(sample)
   power_accu = accumulate(power)
   power_corr = corr[i]*conj(corr[i])
   corr_normal = power_corr / power_accu
end
```



Data dominated kernels in IEEE802.11a synchronization.

Peak detection is control intensive

```
/* LOOP 3 – DETECT (control) */
for i = START:END
  if ((corr_normal[i]>max) and (corr_normal[i] > THRESHOLD)) then
    max = corr_normal
    pos  = i
  end
  if ((corr_normal[i]<max) and (i==pos+TRAILINGSMALLER)) then
    return(pos)
  end
end
```

Peak detection must be sample accurate -> only low potential for vector processing.

Conclusion from algorithmic analysis

- input data is a regular stream of complex sample with 5MHz to 20MHz rate (depends on mode)
- local memory needed to buffer correlation window
- kernels can be merged for stream processing (save dmem, reduce latency)
- two kinds of processing needed: control and data
- all computations within 16bit signed precision
- all division operations can be removed by transformations
- for vector size 2^N shuffling may be avoided
- computational complexity:
  16e – 191 op/sample vs. 11a – 82 op/sample
- small code size -> compiler support not critical

Many common primitives between 11a and 16e !

Application specific optimized instruction-set

Vector instructions (128 bit operands)

vcmul	complex vector mult
vadd, vsub	vector add, sub
vasr, vlsl	shift vector elements right,left
vand, vor	and, or vectors
vtriang, vlevel	vector accumulation
vrotX	rotate vectors X positions
vcon	conjugate complex vector
vreal/vimag	real/imag parts of vector

Generate vector (16bit -> 128bit)

spread	fill vector with scalars
vload+-	load vector from address
pinld	blocking read vector from i/o

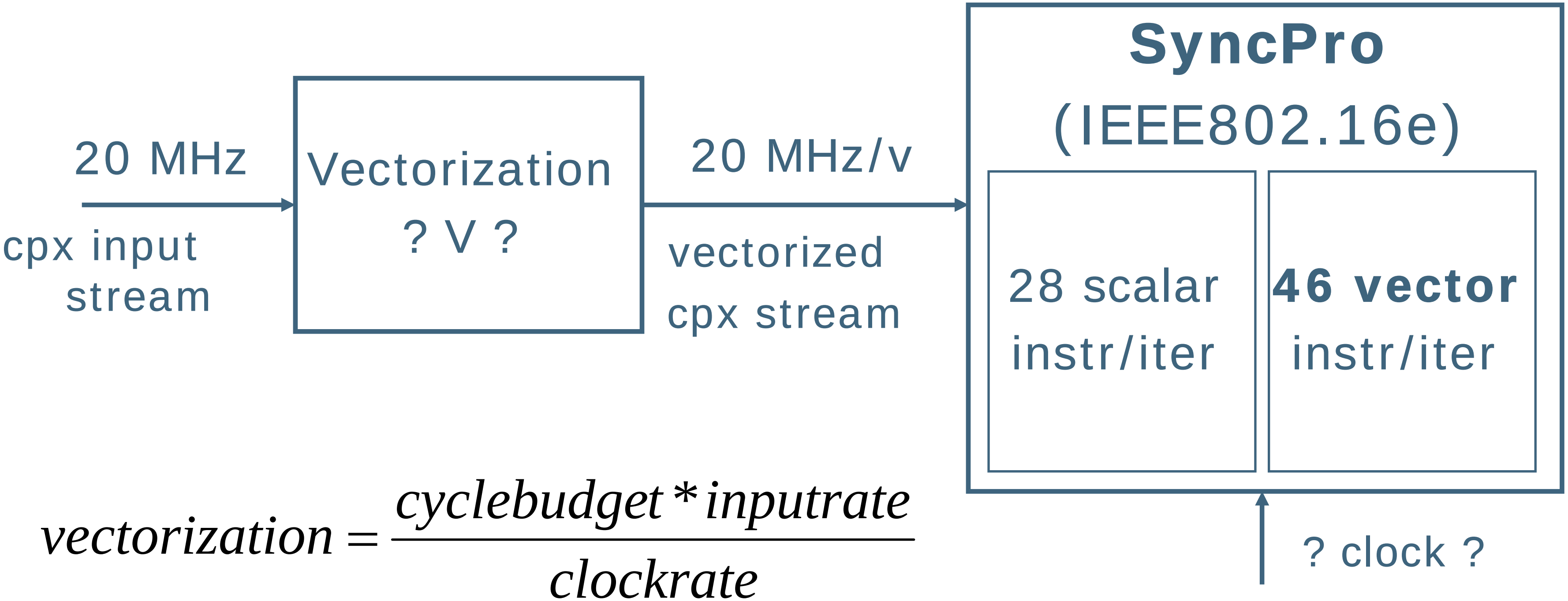
Evaluate vector (128bit -> 16bit)

rgrep/igrep	extract real/imag vector element
rmax/imag	max in real/imag vector elements
vstore+-	store vector to address

Scalar instructions (16bit op)

mov	move imm or reg
mul	multiplication
add	addition
sub	subtraction
asr	arith. shift right
lsl	logic shift left
and	logic and
or	logic or
xor	logic xor
modi	modulo index calc.
pinst	write to i/o
branch	conditional
jump	un-conditional

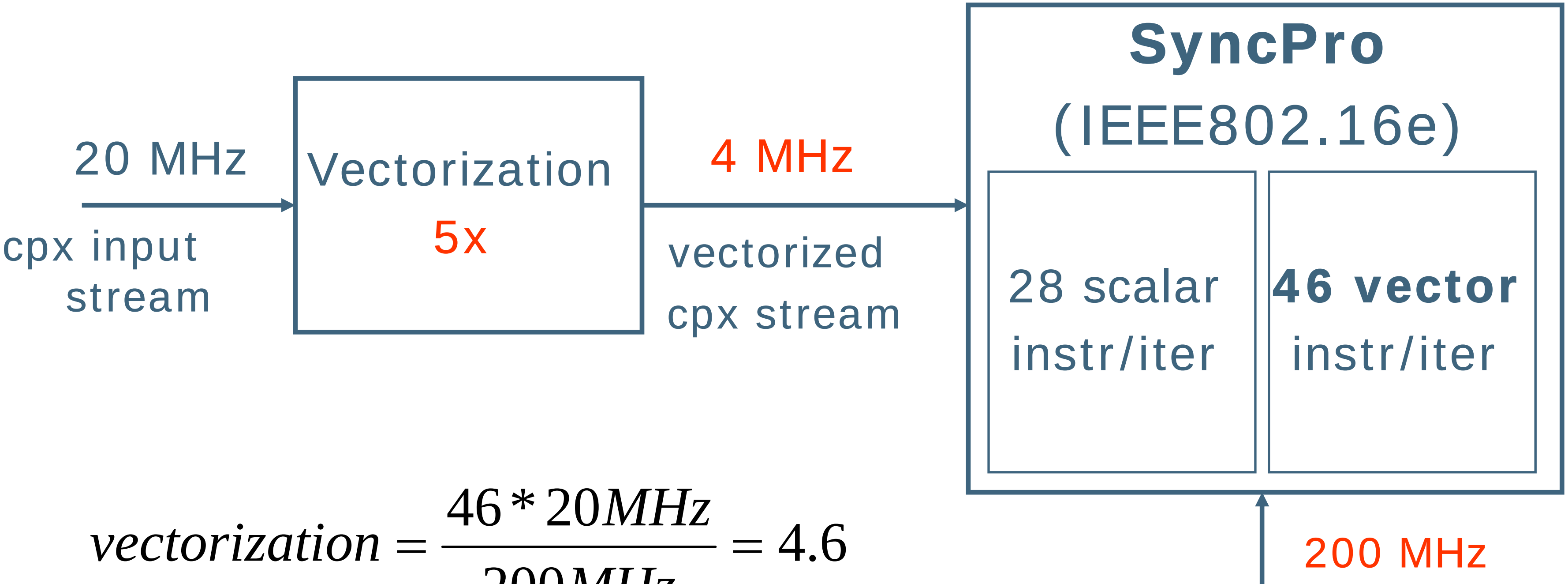
Exploit DLP by vector processing



$$vectorization = \frac{cyclebudget * inputrate}{clockrate}$$

Considered a single vector slot and perfect zero-slack schedule !!

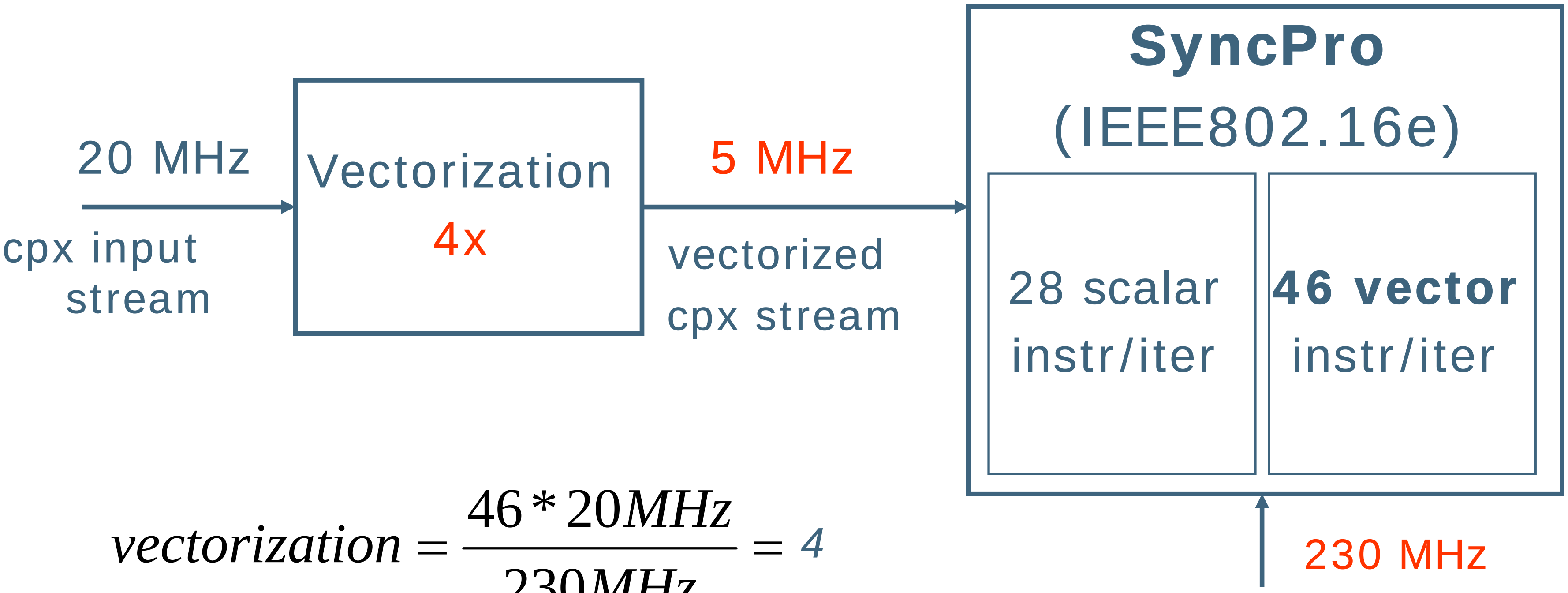
Clock rate should be derived from 200 MHz



$$vectorization = \frac{46 * 20MHz}{200MHz} = 4.6$$

Vector and scalar instructions will be executed in parallel!!

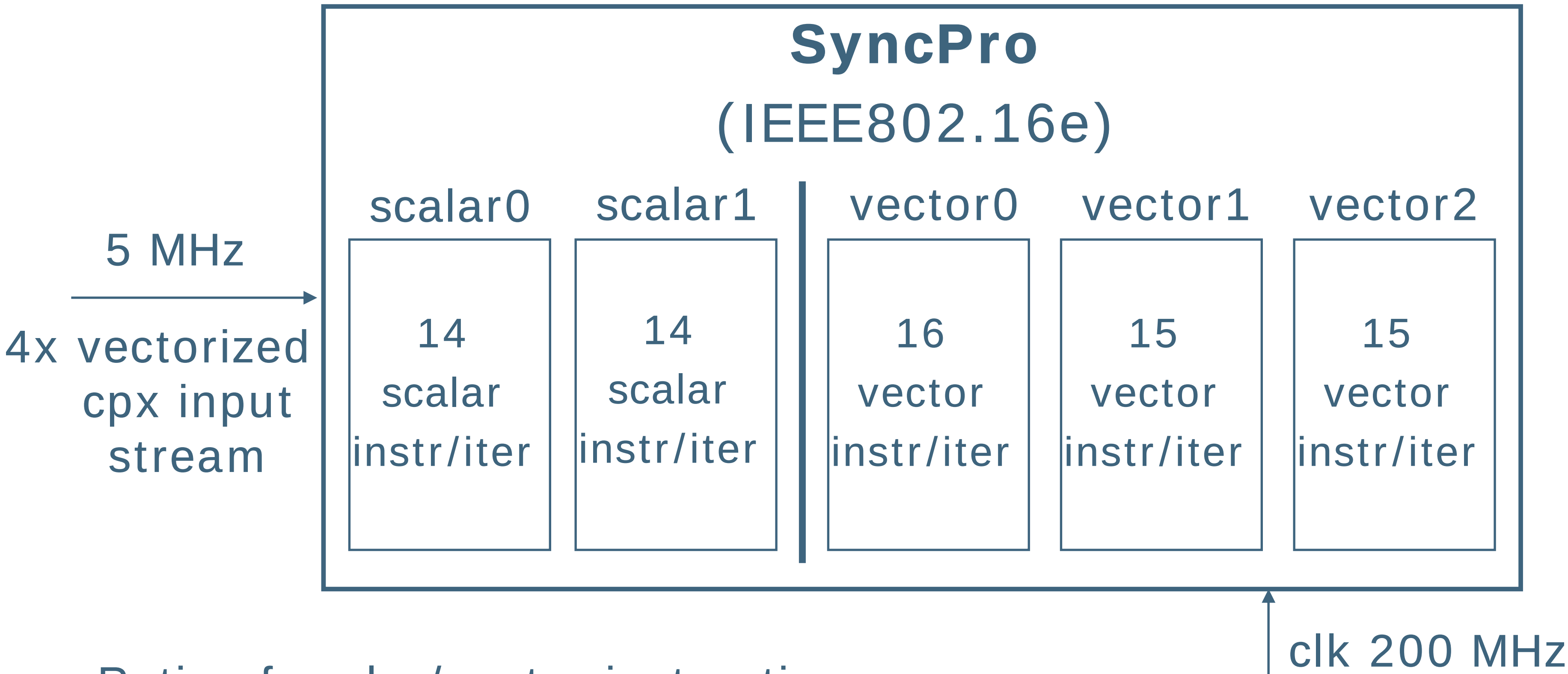
Vectorization should be 2^N



$$vectorization = \frac{46 * 20MHz}{230MHz} = 4$$

- *Clock rates >> 200MHz require deep pipelining.*
- *Vectorization of 8 or 16 does not suit the instruction set.*
  (e.g. area cost vector cmul, long paths in vector accumulation)

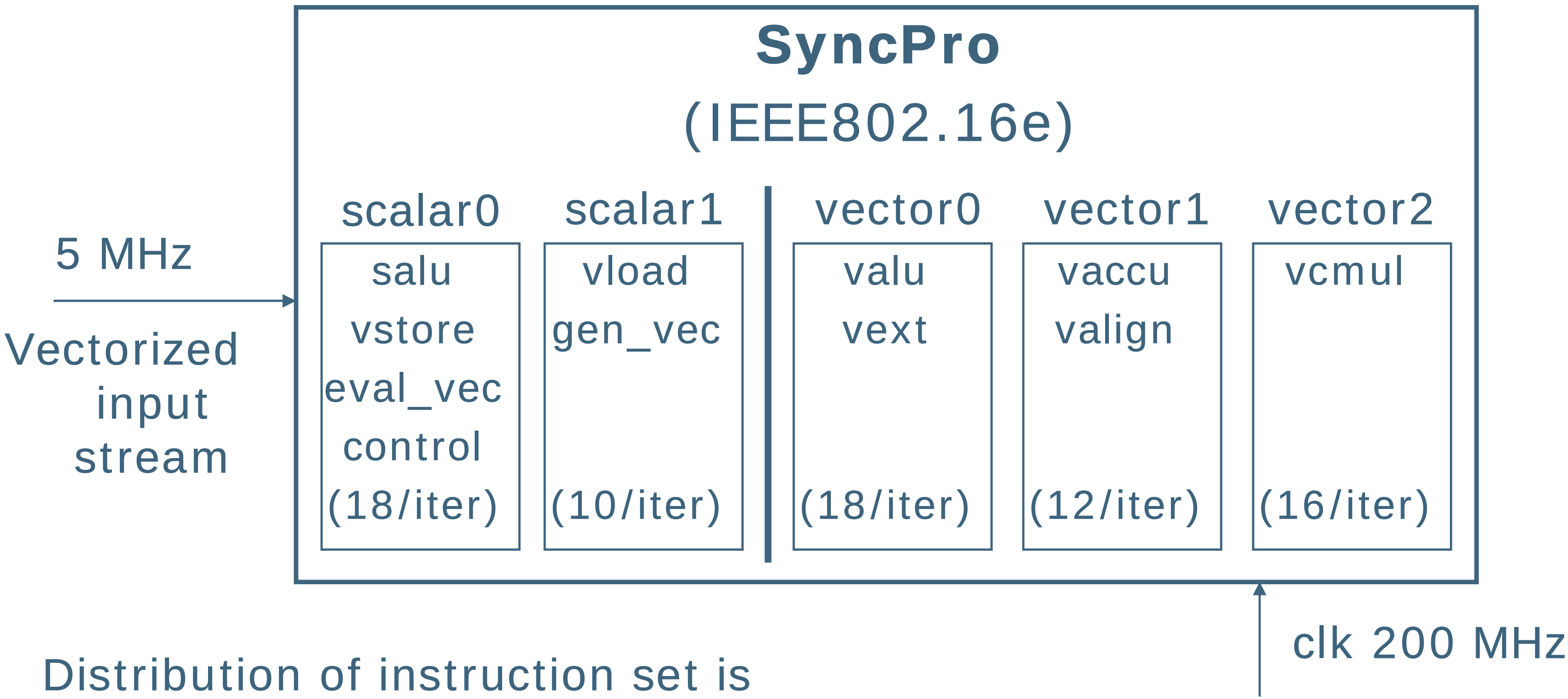
Realistic schedule with 4x vector (200MHz) requires exploitation of ILP



Ratio of scalar/vector instructions (28/46) asks for 2 scalar and 3 vector slots.

$$vec = \frac{16 * 20MHz}{80MHz} = 4$$

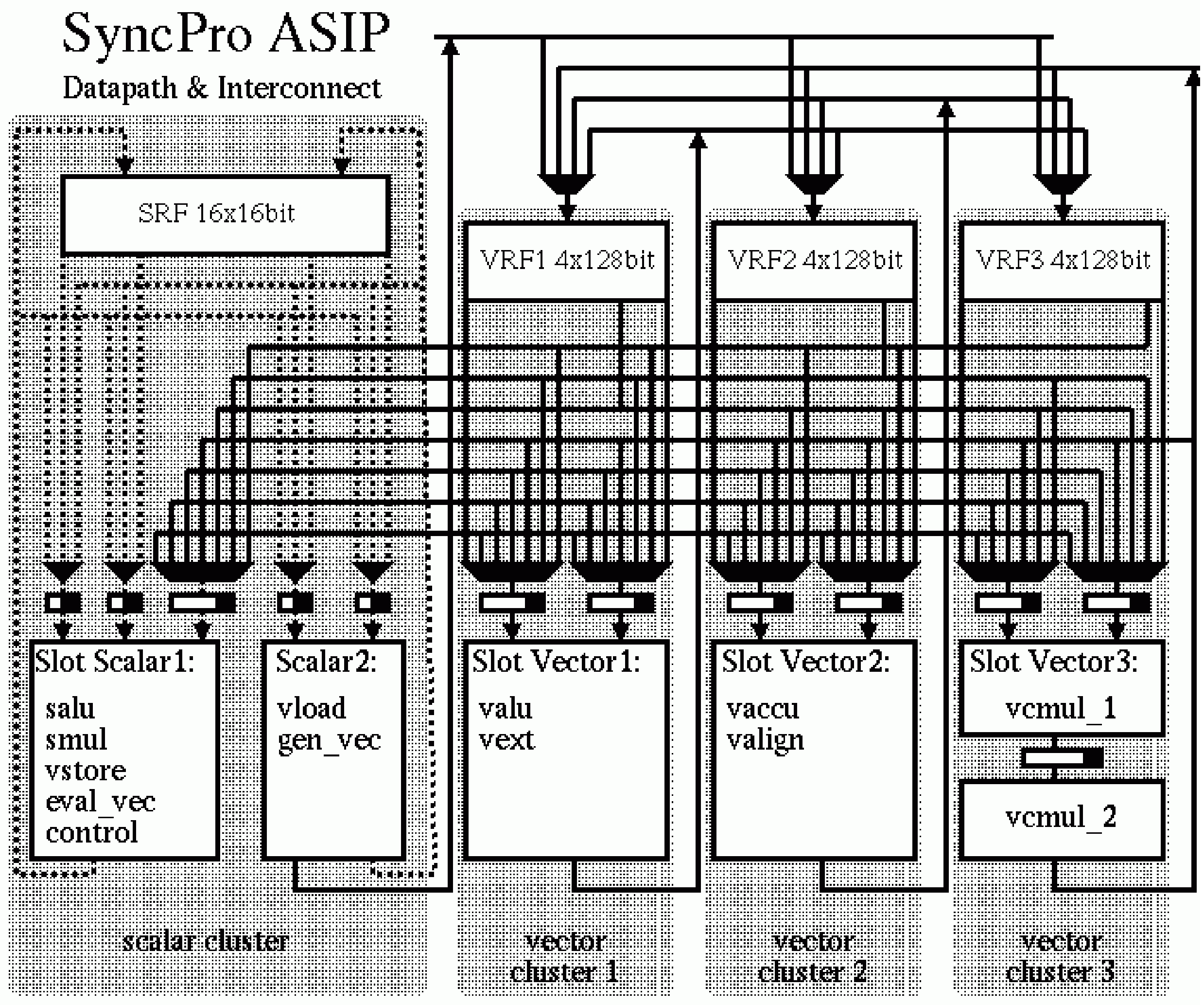
For area and power efficiency instruction set is orthogonalized



Distribution of instruction set is based on instruction statistic, clustering and interconnect !!

$$vec = \frac{18 * 20MHz}{90MHz} = 4$$

Clustering and Interconnect



Features:

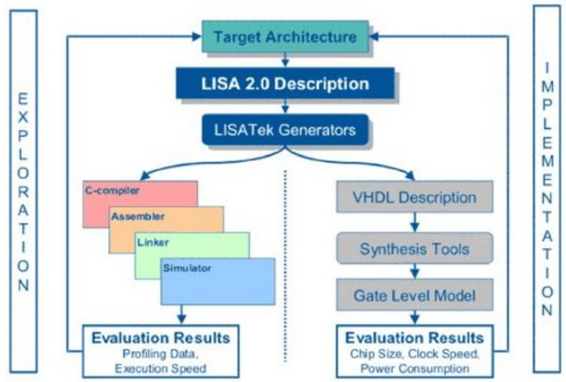
- 5 issue slots
- 1 scalar/3 vector clusters
- SRF 4rp/2wp
- VRF 2rp/1bp/1wp
- scalar units can evaluate and generate vectors
- flexible read/write interconnect

Patent filed

Architecture is modeled with LISATek

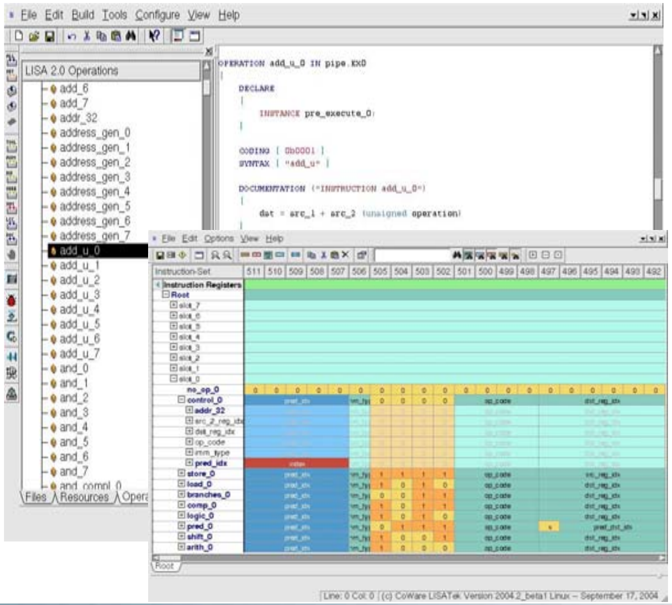
Motivation:

- Automatic software tool generation
- Support for SystemC
  Virtual Platform integration
- Good-quality RTL code generation



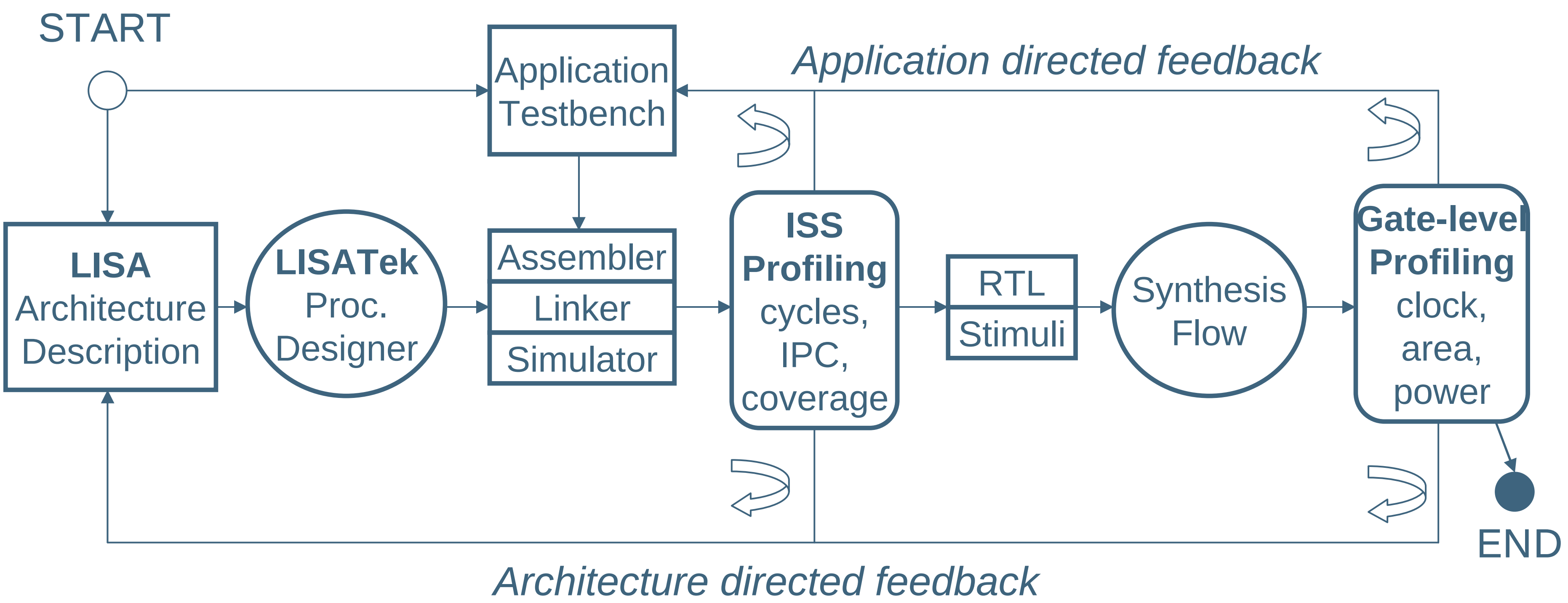
Without tool support development effort is too high.

LISATek gives us time for Exploration!



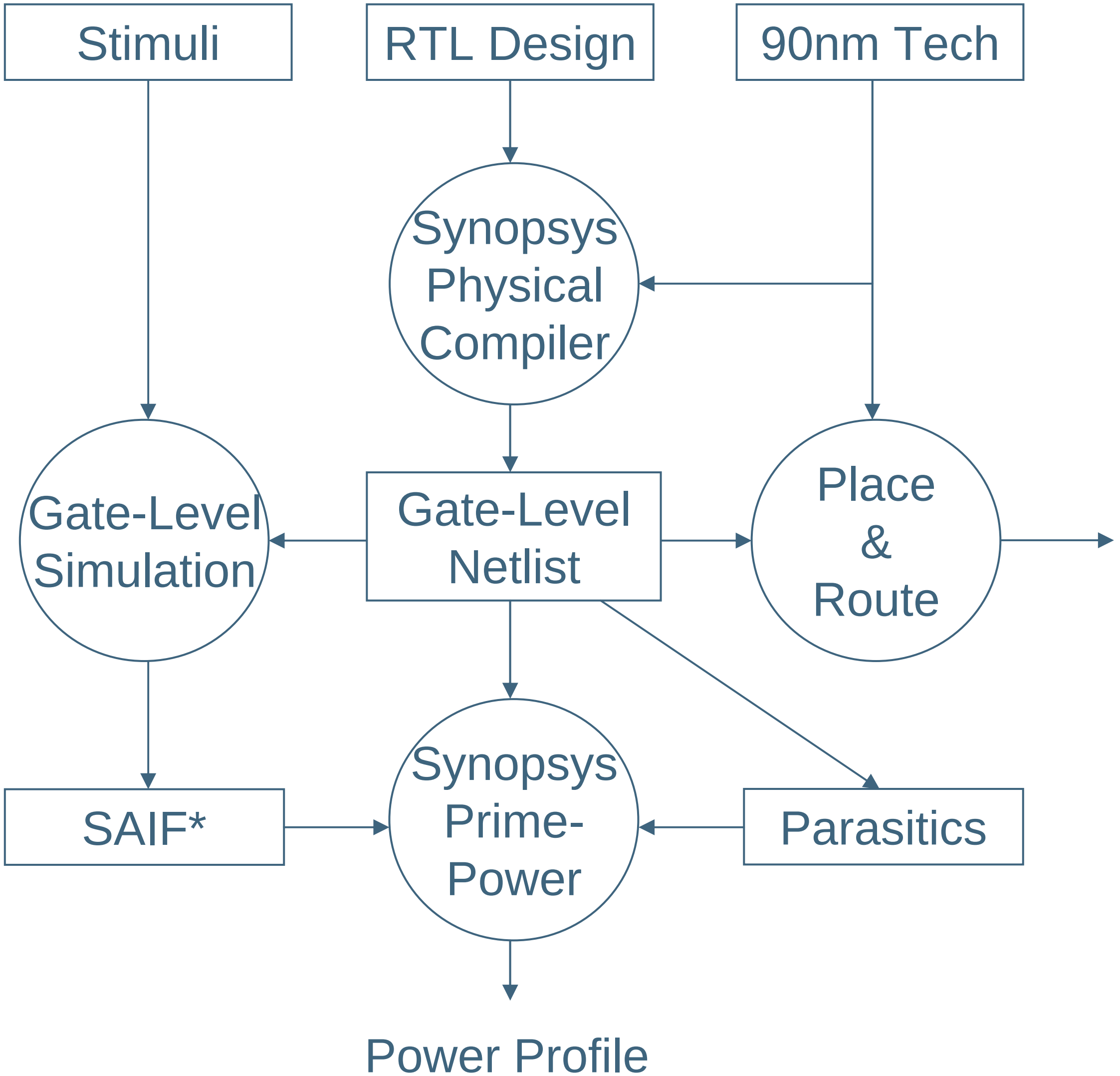
CoWare®
The ESL Design Leader

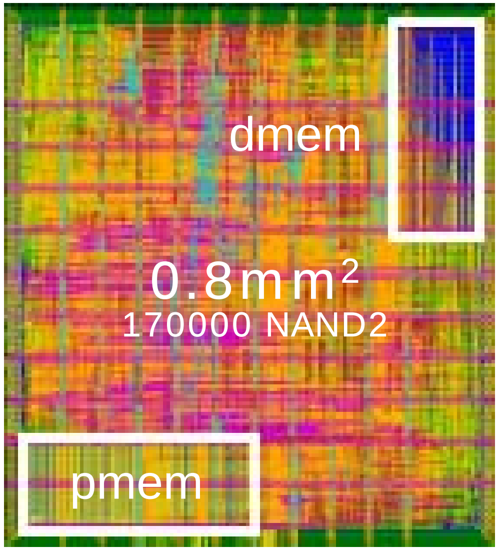
Design and Exploration flow



Efficient implementation requires iterative refinement.

Synthesis & Power Estimation



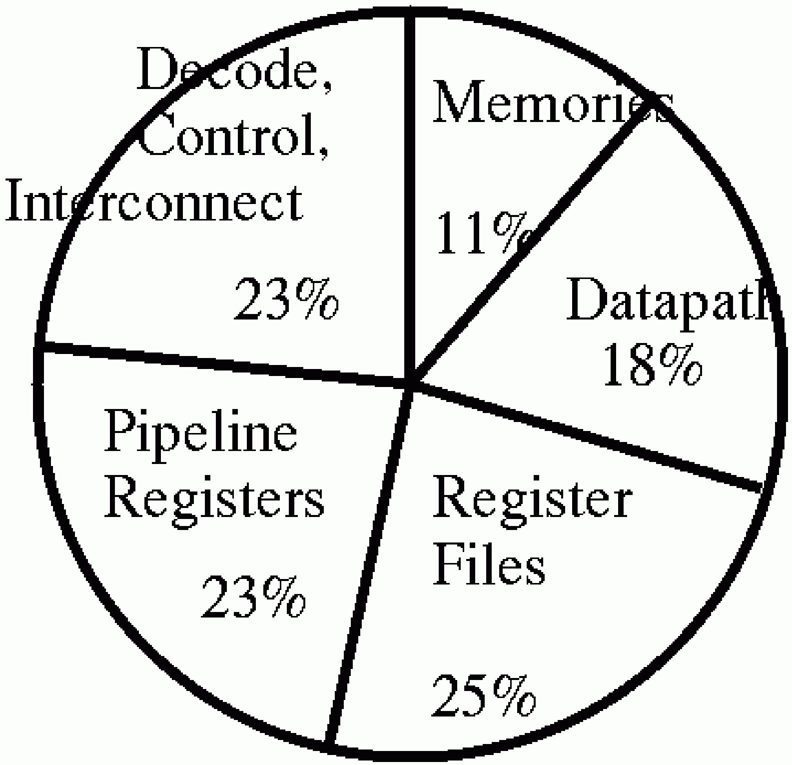


SyncPro2 layout

post-layout timing met (200MHz) !!

*Switching Activity Interface Format

Power Breakdown 802.11a synchronization



power breakdown

optimized design (IEEE802.11a)

- Power consumption is dominated by registers
- Very wide pipeline registers consume 23%
- Low potential for improvement in memories and datapath.

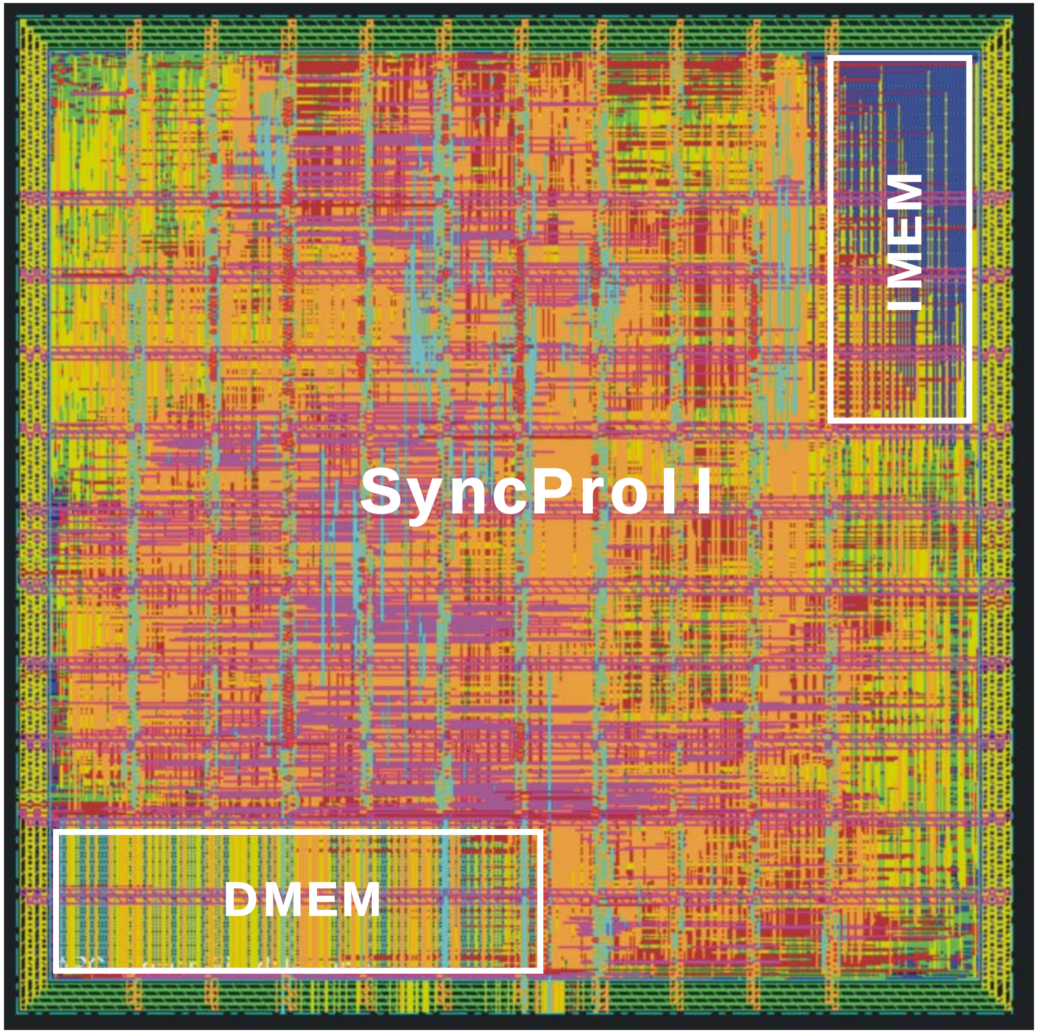
(90nm CMOS 25C, 1V)

SyncPro II Gate-level Design finalized

- Synthesized for TSMC90G with TSMC NVT lib
- Up to 200 MHz WCC
- max performance: 5GOPS (16bit)
- Fine-grain clock-gating based power management

- Area 0.7 sqmm
- Power 18 mW
- >200MOPS/mW (eq. 32b)

- 802.11a/g/n sync @90MHz OK
- 802.16e sync @200MHz OK
- 3GPP-LTE acquisition under development



FLAI platform targets

Targeted standards:

- 802.11n (SDM + Channel bonding) [216Mbps]
- 3GPP-LTE (20MHz BW)
- 802.16e

Area target:

	Area [sqmm]
ARM	**0.7**
BE (2x)	**2 x 6**
DFE (incl. MEM)	**3 x 0.7**
FEC (IP block)	**2 x 0.47**
MEM	**6**
Peripherals	**2**
TOTAL (11n 2x2 20Mhz)	**16 sqmm**
TOTAL (11n 2x2 40Mhz)	**24 sqmm**

TI 3G BB: 10 sqmm/mode
Infineon MUSIC2: ~40sqmm

Platform Power targets:

	WLAN-MIMO	3GPP LTE	UMTS
Idle	3 mW	2mW	7mW
Active max	300 mW	200mW	600mW

Freescale 802.11a/b/g BB:
132mW idle; 200 mW Tx/Rx
Infineon MUSIC2 WCDMA: 350mW

90nm CMOS

aspire invent achieve