CHAIR FOR COMPILER CONSTRUCTION

cfaed CENTER FOR ADVANCING ELECTRONICS DRESDEN

Compiling for deeply embedded and heterogeneous signal processing systems

Jeronimo Castrillon

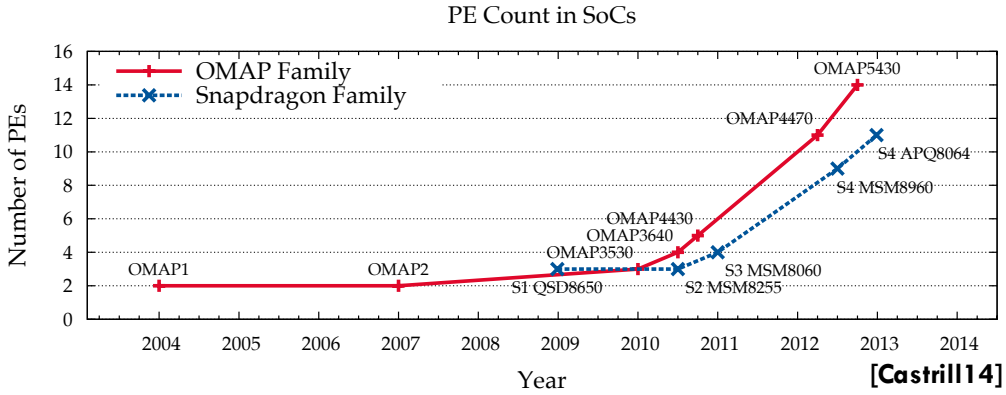
Cfaed Chair for Compiler Construction (CCC)

5G Summit, Dresden, Germany

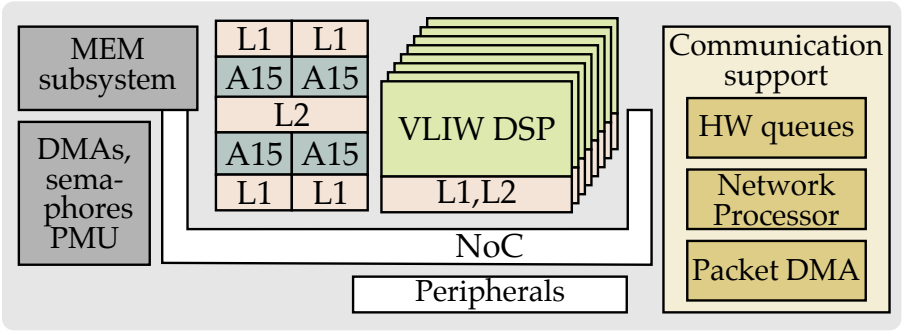
September 29, 2016

Multi-Processor/core Systems on Chip

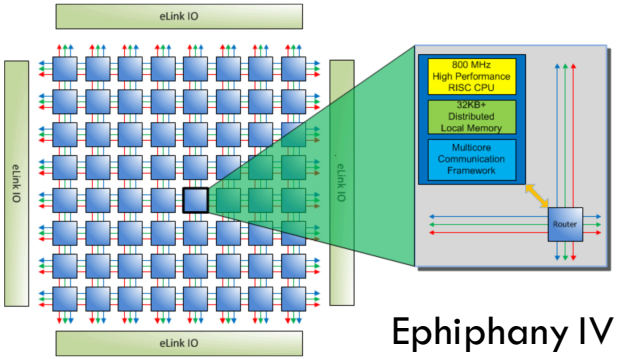
- **HW complexity**
  - Increasing number of cores
  - Increasing heterogeneity
- **Heterogeneity and size**



[Castrill14]



TI Keystone II



Ephiphany IV

Source: http://www.adapteva.com/docs/e64g401_datasheet.pdf

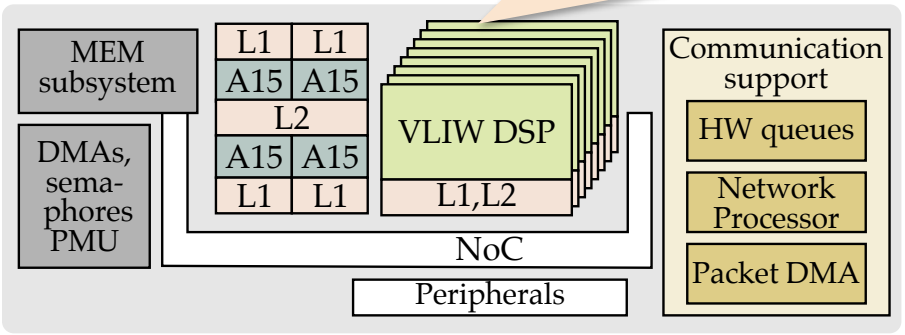
Multi-Processor/core Systems on Chip

- **SW productivity gap**
  - How to program these systems?
  - Meet performance/energy requirements

5G

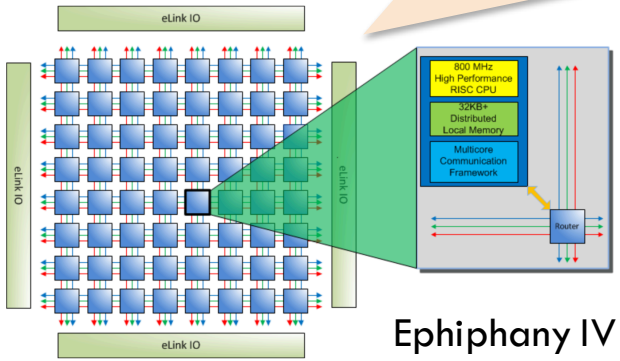
Massive MIMO | High-perf. Codes | Complex Beamform.

Fragmented tools, different runtimes/OS on ARM and DSP, 500+ pages on APIs



TI Keystone II

Homogeneous! OpenMP support uses 1/3 of program memory!



Ephiphany IV

Source: http://www.adapteva.com/docs/e64g401_datasheet.pdf

Multi-Processor/core Systems on Chip

- SW productivity gap
  - How to program these systems?
  - Meet performance/energy requirements

5G

Massive MIMO | High-perf. Codes | Complex Beamform.

Fragmented tools, different runtimes/OS on ARM and DSP, 500+ pages on APIs

TI Keystone II

Homogeneous! OpenMP support uses 1/3 of program memory!

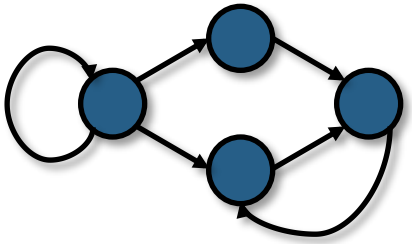
hiphany IV

Source: http://www.adapteva.com/docs/e64g401_datasheet.pdf

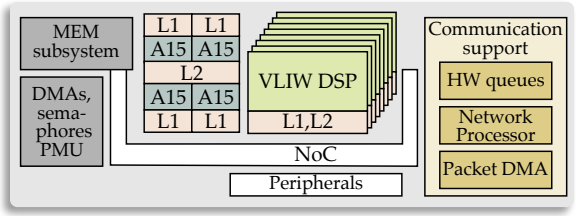
Need for software automation tools

Programming flow

Dataflow application



Architecture model



Non-functional specification

Analysis

Synthesis

Code generation

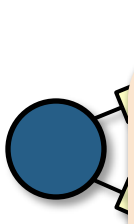
[Castrill14]

Property models (timing, energy, error, …)

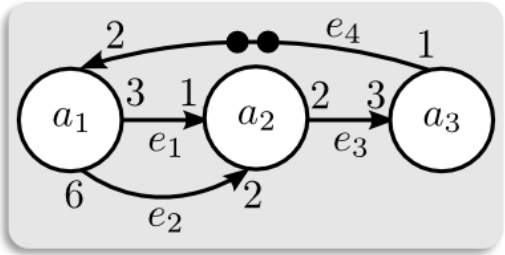
```
PNargs_ifft_r.ID  = 6U;
    PNargs_ifft_r.PNchannel_freq_coef  = f
    PNargs_ifft_r.PNnum_freq_coef  = 0U;
    PNargs_ifft_r.PNchannel_time_coef  = s
    PNargs_ifft_r.channel  = 1;
    sink_left  = IPCllmrf_open(3,  1, 1);
    sink_right  = IPCllmrf_open(7,  1, 1);
    PNargs_sink.ID  = 7U;
    PNargs_sink.PNchannel_in_left  = sink_l
    PNargs_sink.PNnum_in_left  = 0U;
    PNargs_sink.PNchannel_in_right  = sink_
    PNargs_sink.PNnum_in_right  = 0U;
    taskParams.arg0  = (xdc_UArg)&PNargs_src
    taskParams.priority  = 1;

ti_sysbios_knl_Task_create((ti_sysbios_knl_
&taskParams,  &eb);
    glob_proc_cnt++;
    hasProcess  = 1;
    taskParams.arg0  = (xdc_UArg)&PNargs_fft
    taskParams.priority  = 1;

ti_sysbios_knl_Task_create((ti_sysbios_knl_
ft_Templ,  &taskParams,  &eb);
    glob_proc_cnt++;
    hasProcess  = 1;
    taskParams.arg0  = (xdc_UArg)&PNargs_ift
    taskParams.priority  = 1;

ti_sysbios_knl_Task_create((ti_sysbios_knl_
fft_Templ,  &taskParams,  &eb);
    glob_proc_cnt++;
    hasProcess  = 1;
    taskParams.arg0  = (xdc_UArg)&PNargs_sin
    taskParams.priority  = 1;
```

Dataflow programming models

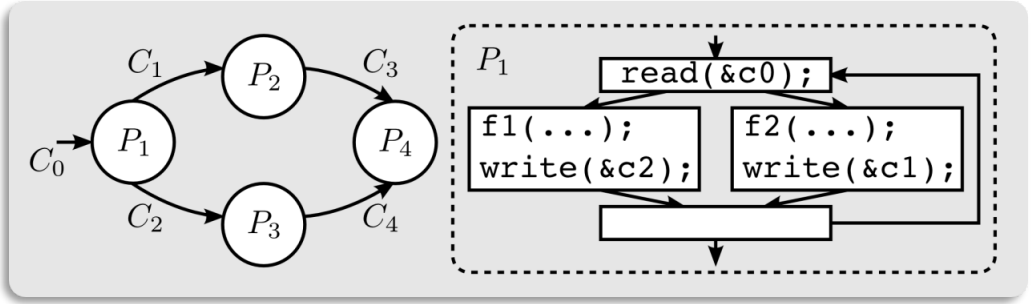
- Graph representation of applications
  - Implicit repetitive execution of tasks
  - Good model for streaming applications
  - Good match for signal processing & multi-media applications
- Large body of research on multiple flavors of these models



Properties: No race conditions, determinism, strong/weak guarantees



Static



Dynamic

Language: C for process networks

- FIFO Channels

```
typedef struct { int i; double d; } my_struct_t;
__PNchannel my_struct_t S;
__PNchannel int A = {1, 2, 3}; /* Initialization */
__PNchannel short C[2], D[2], F[2], G[2];
```

- Processes & networks

```
__PNkpn AudioAmp __PNin(short A[2]) __PNout(short B[2])
                 __PNparam(short boost){
  while (1)
    __PNin(A) __PNout(B) {
      for (int i = 0; i < 2; i++)
        B[i] = A[i]*boost;
    }}
__PNprocess Amp1 = AudioAmp __PNin(C) __PNout(F) __PNparam(3);
__PNprocess Amp2 = AudioAmp __PNin(D) __PNout(G) __PNparam(10);
```

[Sheng14]

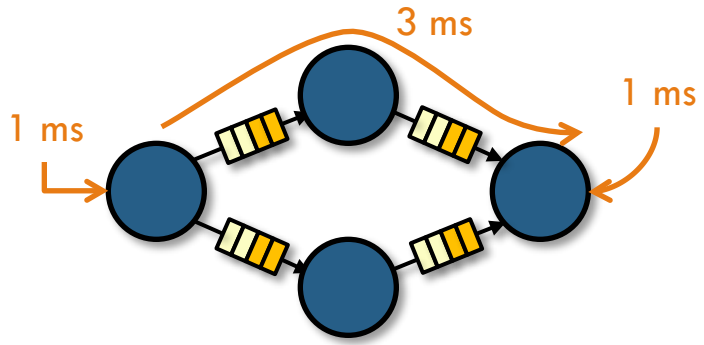
Architecture model and constraints

- System model including:
  - Topology, interconnect, memories
  - Computation: uArch model
  - Communication: cost functions
  - MCA standardizing it to SHIM 2.0

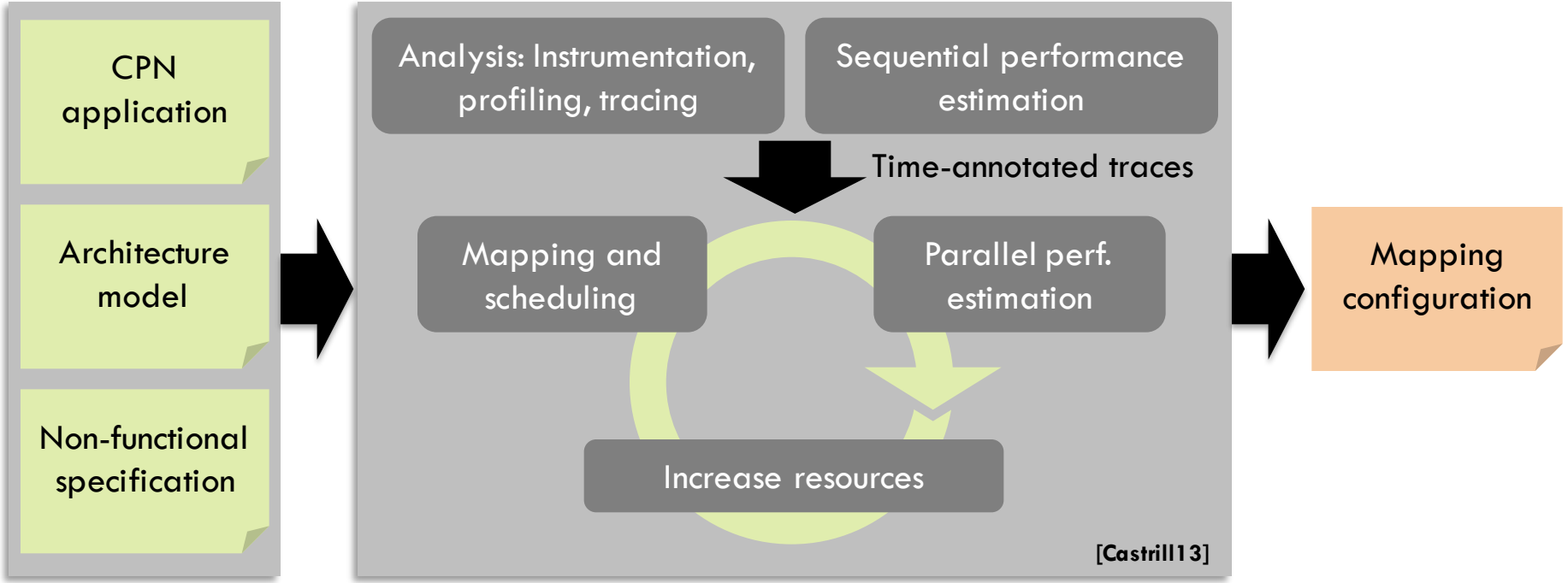
```
-<Platform>
   <Processors List="dsp0 dsp1 dsp2 dsp3 dsp4 dsp5 dsp6 dsp7"/>
   <Memories List="local_mem_dsp0_L2 local_mem_dsp1_L2 local_mem_dsp2_L2 local_mem_dsp3_L2 local_mem_d
   local_smem_dsp1_L2 local_smem_dsp2_L2 local_smem_dsp3_L2 local_smem_dsp4_L2 local_smem_dsp5_L2 local_s
   local_mem_dsp3_DDR local_mem_dsp4_DDR local_mem_dsp5_DDR local_mem_dsp6_DDR local_mem_dsp7_DDR sh
   <CommPrimitives List="IPCll_SL2 IPCll_DDR EDMA3_SL2 EDMA3_DDR EDMA3_LL2"/>
 </Platform>
 <Processor Name="dsp0" CoreRef="DSPC66"/>
 <Processor Name="dsp1" CoreRef="DSPC66"/>


 <Processor Name="dsp7" CoreRef="DSPC66"/>
-<Memory>
   <LocalMemory Name="local_mem_dsp0_L2" Size="524288" BaseAddress_hex="00800000" ProcessorRef="
 </Memory>
 ...
```

[Oden13]

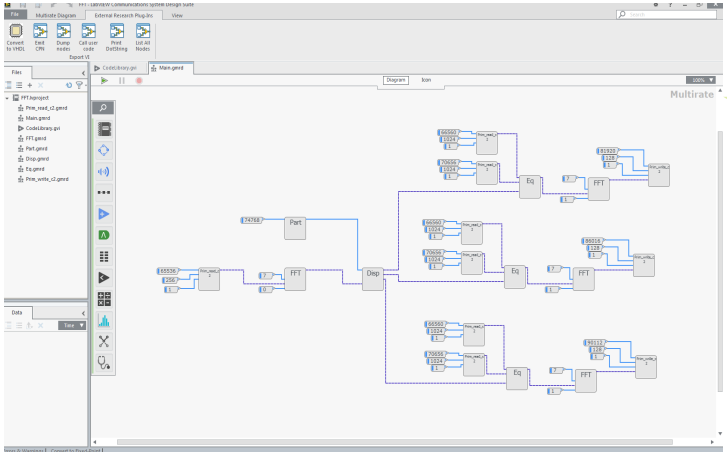
- Constraints
  - **Time constraints**
  - Mapping constraints
  - Platform constraints



Analysis and synthesis: Overview



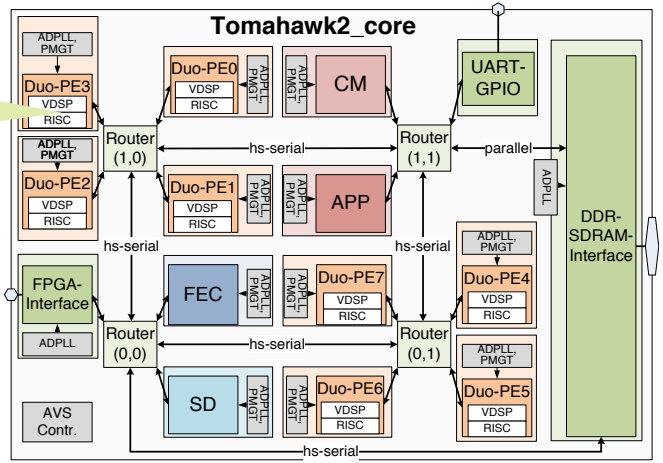
Example 1) From LabVIEW to Tomahawk Platform



Baseband processing application
(data dependent execution paths)

Visit demo booth

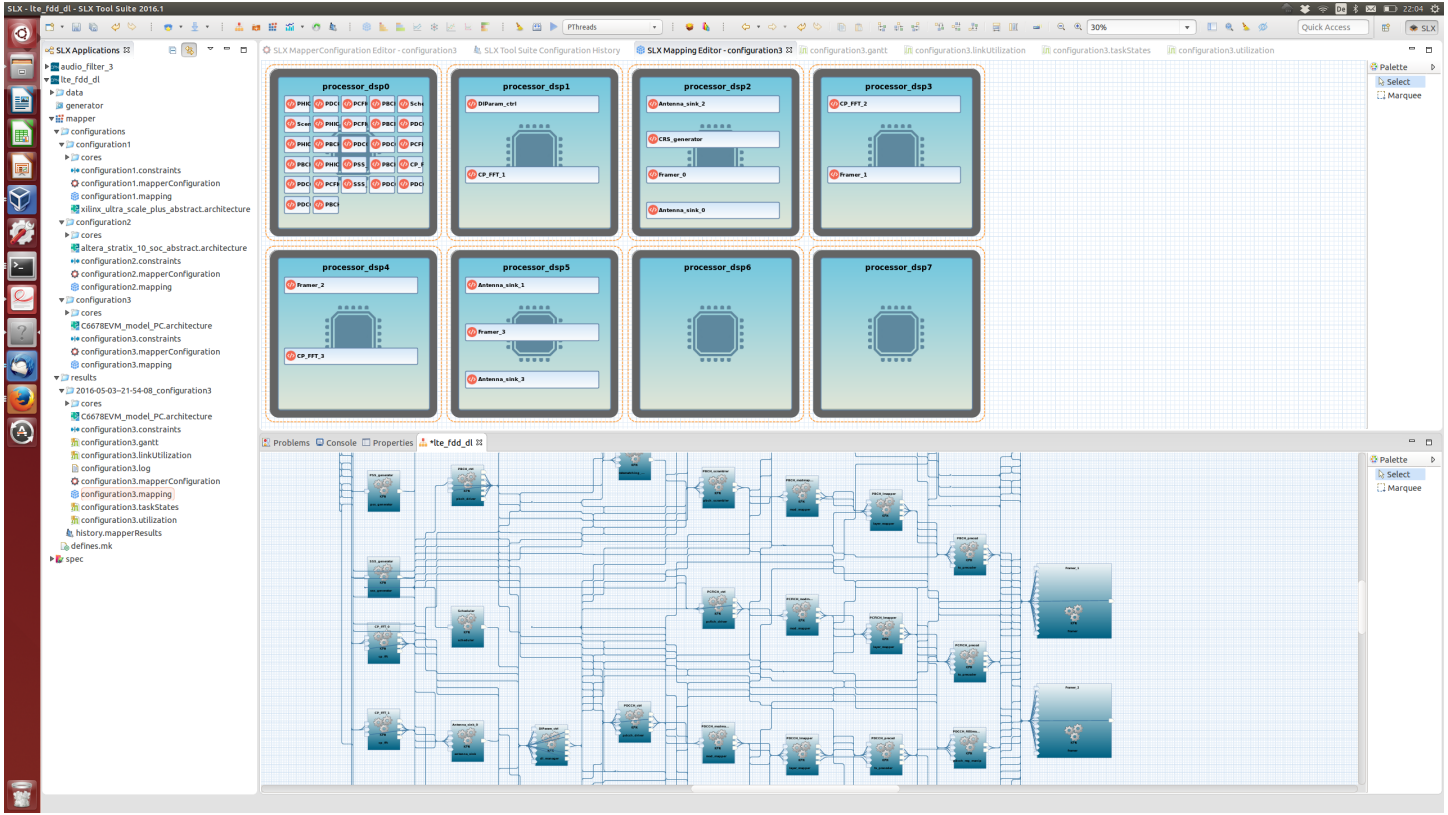
Automatic code generation



[Arnold13]

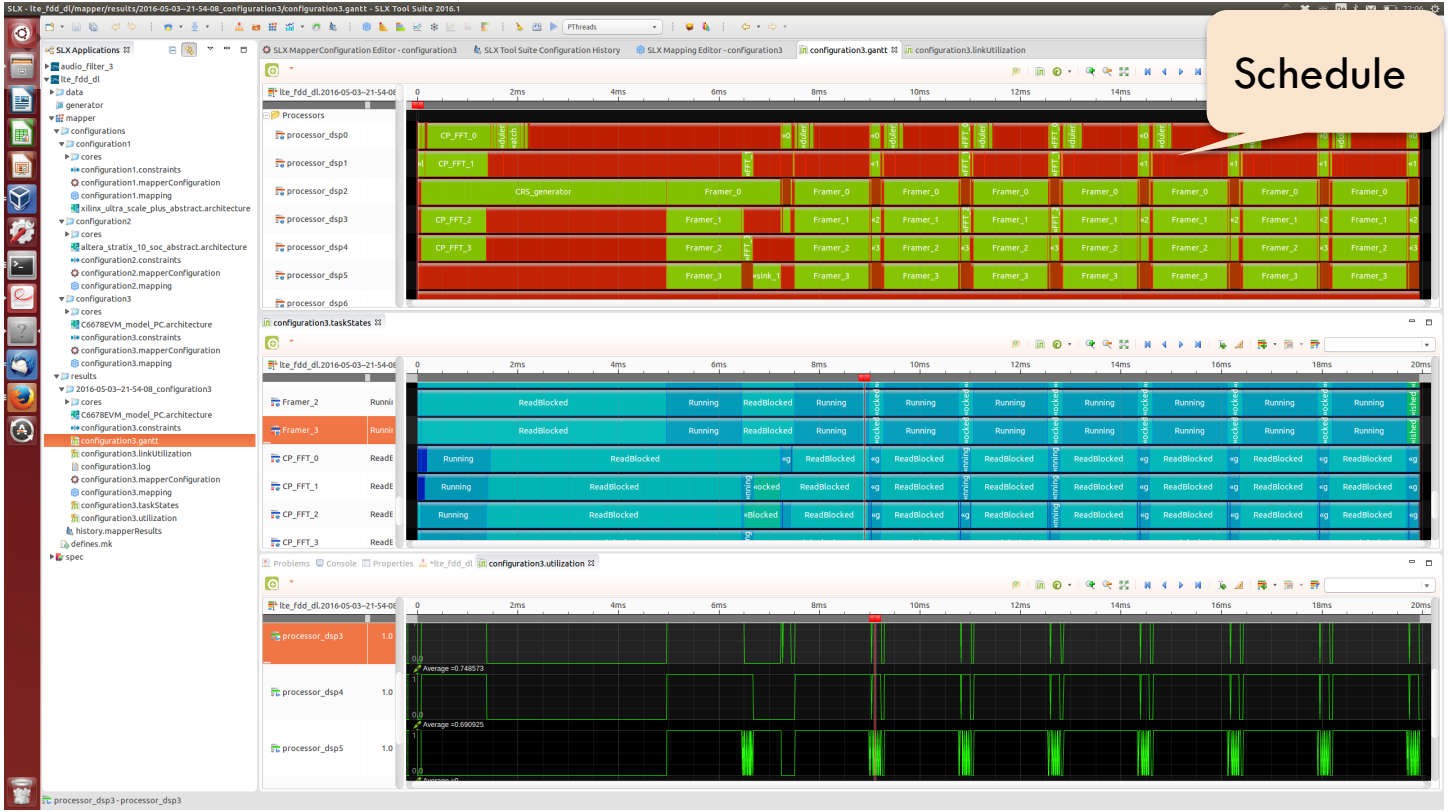
Tomahawk Chip (accelerators for baseband processing)

Example 2) LTE application mapping



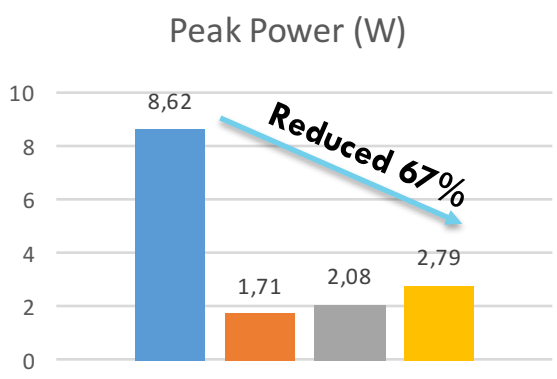
Cortessy: Silexica Software Solutions GmbH

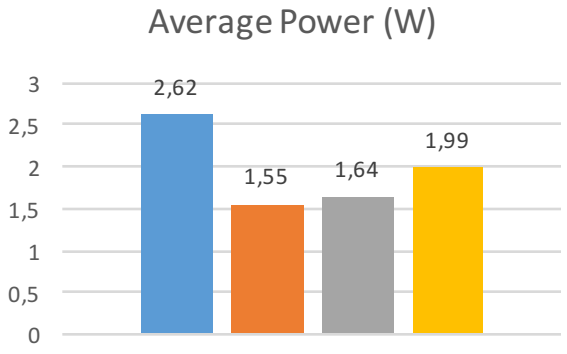
Example 2) LTE application execution

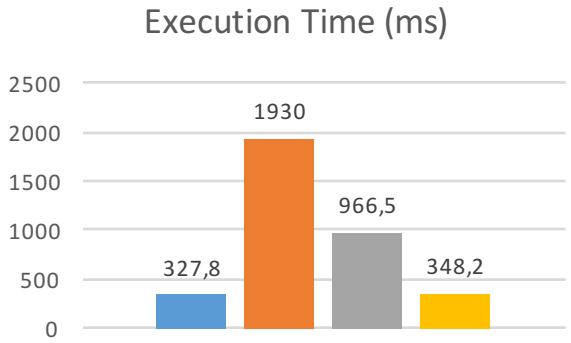


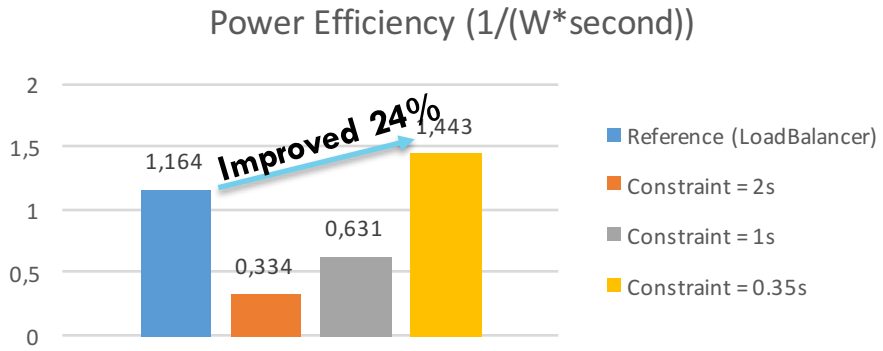
Cortessy: Silexica Software Solutions GmbH

Example 2) LTE application: Results









Cortessy: Silexica Software Solutions GmbH

Acknowledgements

- Vodafone Chair for Mobile Communications Systems
- Silexica Software Solutions GmbH
- National Instruments
- German Cluster of Excellence: Center for Advancing Electronics Dresden (www.cfaed.tu-dresden.de)

Thanks!
Questions?

References

[**Castrill14**] J. Castrillon and R. Leupers, Programming Heterogeneous MPSoCs: Tool Flows to Close the Software Productivity Gap. Springer, 2014

[**Sheng14**] W. Sheng, S. Schürmans, M. Odendahl, M. Bertsch, V. Volevach, R. Leupers, and G. Ascheid, "A compiler infrastructure for embedded heterogeneous MPSoCs", Parallel Comput. 40, 2 (February 2014), 51-68

[**Oden13**] M. Odendahl, et al., "Split-cost communication model for improved MPSoC application mapping", In International Symposium on System on Chip pp. 1-8, 2013

[**Castrill13**] J. Castrillon, R. Leupers, and G. Ascheid, "MAPS: Mapping concurrent dataflow applications to heterogeneous MPSoCs," IEEE Transactions on Industrial Informatics, vol. 9, no. 1, pp. 527–545, 2013

[**Arnold13**] O. Arnold, et al. "Tomahawk - Parallelism and Heterogeneity in Communications Signal Processing MPSoCs". *TECS, 2013*