Übersetzerbau in österreichischen Softwarefirmen

TU Wien

11.4.2019

CATENA

“Partner of choice in System IP and IC Design”

- IC design-house founded in 1986
- Shares acquired by NXP Semiconductors, April 2012
- Maintains independent operations, working for 3rd parties
- Development of *RF, Analog, Mixed-Signal and DSP Systems* with a focus on *Integrated Circuits* and *Wireless IPs*
- 6 development centers
- http://www.catena.tech

ISO 9001:2000

QualityMasters

SRTCI ACREDITED

Offices

Delft, The Netherlands **1986**

Headquarters

RF/AMS design, RF System architecture

Eindhoven, The Netherlands **2000**

System Architecture,
Digital and DSP design, Embedded SW

Kista, Sweden **2001**

RF/AMS design

RF System architecture

Vienna, Austria **2006**

DSP Architecture and Tooling

Dresden, Germany **2016**

Digital design

Embedded SW

Pavia, Italy **2017**

RF/AMS design

Applications Areas

- **Wireless Communication**
  - Datacom (Bluetooth, WiFi 802.11x, WiMax, ZigBee, ISM Bands)
  - Proprietary Wireless Links (e.g.: Wireless Loudspeakers)
  - RFID Readers
- **Consumer & Automotive Electronics**
  - FM/AM Broadcast (Car, Portable and Mobile Phones)
  - Digital Radio (DAB(+), DMB-T, DRM(+), RDS, DARC)
  - TV Tuners (DVB Terrestrial, Cable and Satellite, Mobile TV)
  - GPS, GLONASS, Beidou Front-End
  - Keyless Entry and Immobilizer
  - Sensor Electronics
- **Industrial & Medical**
  - Advanced Measurement Circuits (MEMS Gyro/Accelerometer)
  - Implantable Wireless Communication (Pacemakers)
  - Electron Beam Deflector (deep submicron manufacturing equipment)

Catena DSP GmbH

Albertgasse 35, 1080 Wien
4 employees

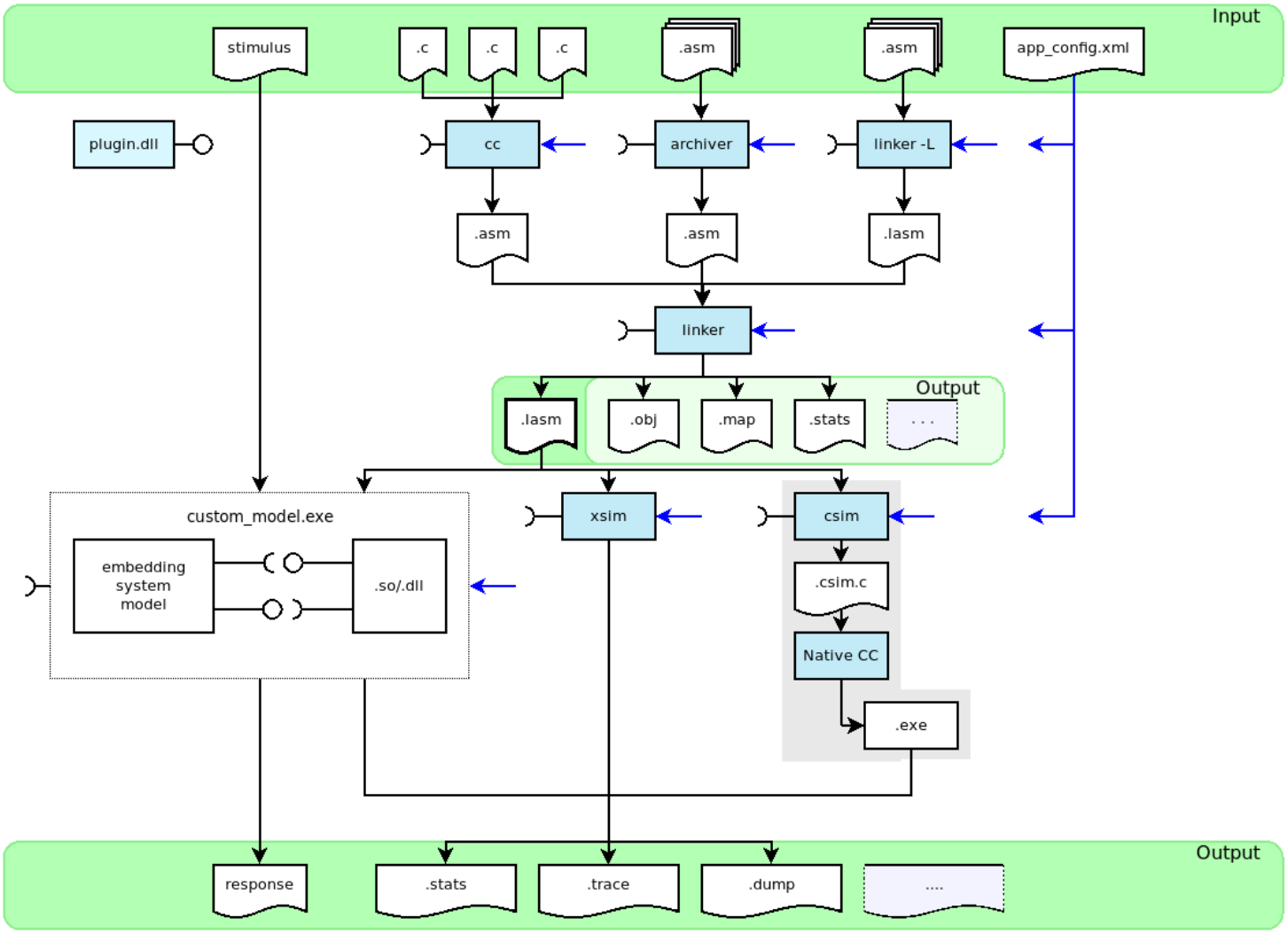
⇨ TEFEC

***T**OOLING **E**NVIRONMENT **F**OR **E**MBEDDED **C**ORES*

⇨ AAA

***A**PPLICATION-SPECIFIC **A**DAPTABLE **A**RCHITECTURE*

Overview


Input
stimulus
.c
.c
.c
.asm
.asm
app_config.xml
plugin.dll
cc
archiver
linker -L
.asm
.asm
.lasm
linker
Output
.lasm
.obj
.map
.stats
. . .
custom_model.exe
embedding system model
.so/.dll
xsim
csim
.csim.c
Native CC
.exe
Output
response
.stats
.trace
.dump
....

CATENA

Design space exploration

- Goal
  - Optimize chip area
  - Reduce power dissipation & energy consumption
- Based on a central architecture configuration
- Tools are dynamically configured to a specific target processor
- 2 phases
  - Evaluation phase
  - Production phase

CATENA

Evaluation

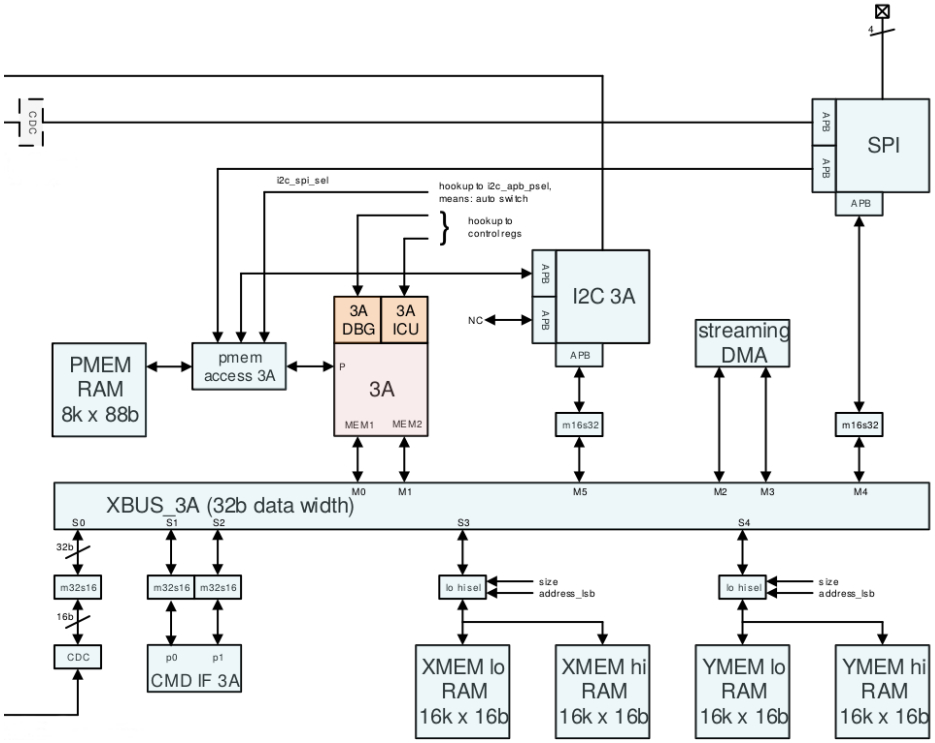
- Find out important key parameters
  - Number of registers
  - VLIW width
  - Custom instructions
- Iterative approach
  - Build
  - Simulate
  - Evaluate
  - Modify
  - Repeat until happy

Production

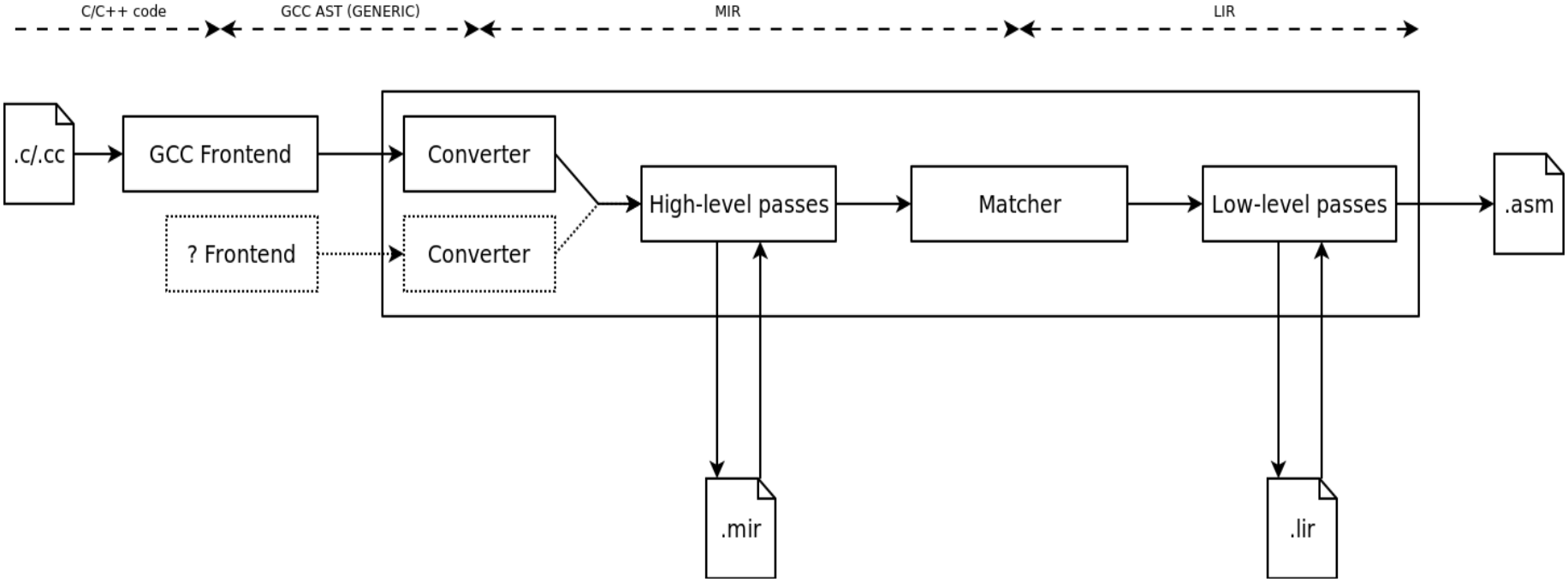
- Key parameters settled
  - Fine tuning
  - Binary encoding
- Generate HW description
  - e.g. instruction decoder
  - Verilog, VHDL

Embedded Core - Example

AAA



Compiler overview



CATENA

Target independent MIR passes

- SSA
- Inlining
- Loop optimizations
- SIMD code generation
- Copy- and constant-propagation
- Partial redundancy elimination
- Hardware loops
- Instruction selection

CATENA

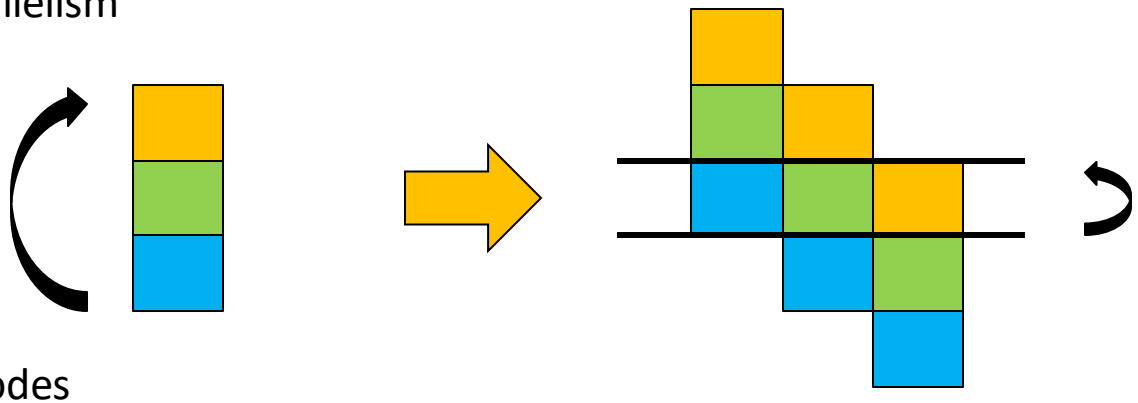
Target dependent LIR passes

- Control-flow optimizations
- IF-conversion
- VLIW instruction scheduling
- Software pipelining
- Register allocation
- Coalescing
- Spilling

domain specifics (1)

- Processor (micro) architecture is scalable in various key parameters
- Harvard processor architectures

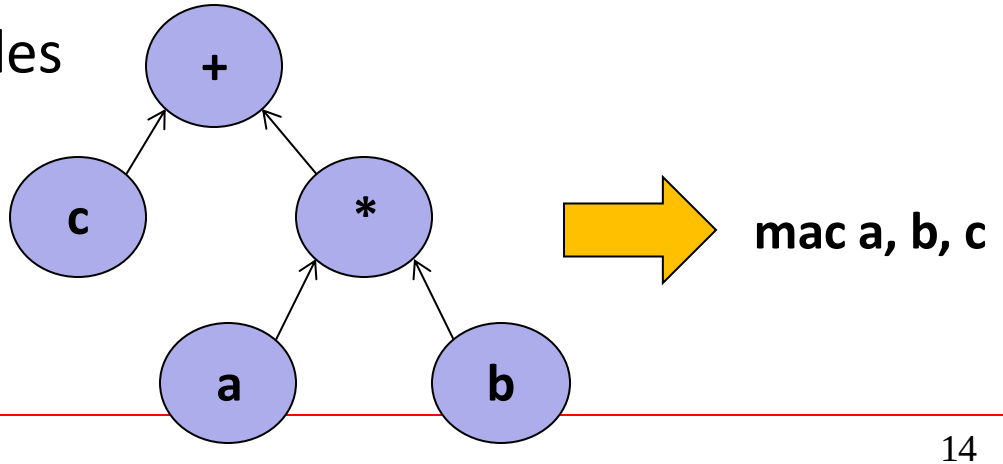
- VLIW ISA, explicit parallelism
  - Scheduler, SWP



- Special Addressing Modes
  - Support for code like “x = *p++;”
- Fixed point calculations
  - Type system extensions
  - Specialized code recognized during tree pattern matching

domain specifics (2)

- Delayed branches
  - Explicit delay slots
- Predicated Execution
  - IF-conversion
- Hardware Loops
  - Used for loops with statically known iteration count
- DSP Instructions, e.g. multiply-accumulate
  - Pattern matching rules



Example: FIR filter kernel

fir_kernel.c

```
#define FILTER_ORDER  26
#define N             (FILTER_ORDER + 1)

int fir_kernel(int delay[N], int coeff[N])
{
    long a = 0;
    for (int i = 0; i < N; ++i)
        a += ((long)delay[i] * coeff[i]) << 1;

    return a >> 16;
}
```

```
fir_kernel:

b0:
    mov.i 0, L0
    ld (R0) += 1, hD1 || ld (R1) += 1, lD1 || bkrep.i 25, b2_end
    ld (R0) += 1, hD1 || ld (R1) += 1, lD1
b2:
    ld (R1) += 1, lD1 || ld (R0) += 1, hD1 || mac.f hD1, lD1, A0
b2_end:
b5:
    mac.f hD1, lD1, A0
    mac.f hD1, lD1, A0
    ret
    mov hD0, lD0
    mnop
```

Finally …

- Internship/master's thesis
  - On request
- We are looking for an embedded firmware developer

  write an email to
  - christian.panis@catena.tech
  - ulrich.hirnschrott@catena.tech
  - benedikt.lukas.huber@catena.tech (that is me)
- Questions?