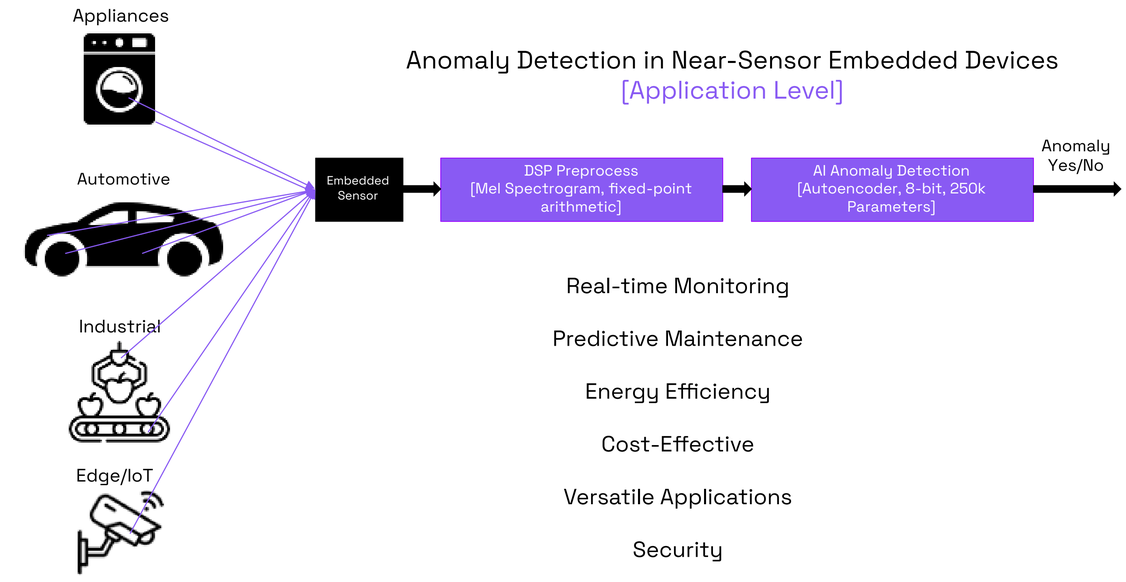
Optimizing neural network classification on resource constrained processors through customization

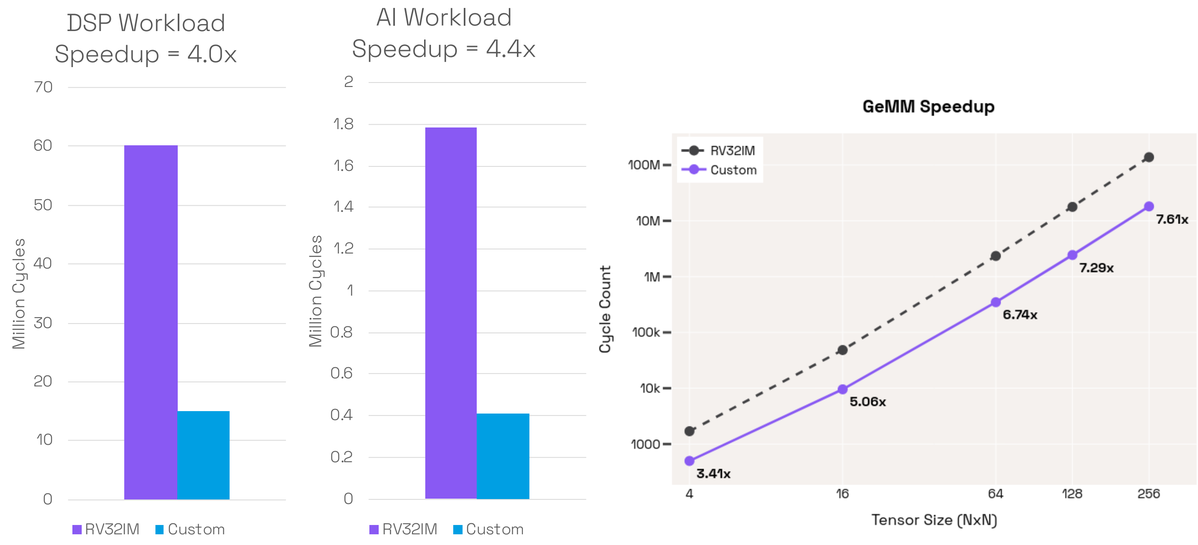
Keith Graham, Tadej Murovič

Codasip

→ AI/DSP on tiny processors optimized through bounded customization



→ Performance



→ Align the Software / Hardware Interface

- Add Fixed-Point DSP instructions without the full overhead of the floating-point F-extension
- Add 8-bit SIMD instructions for AI workloads.
- Add Single cycle loads. Load next word, align & pack with previous load. Update address pointer.
- Enabling the core to run at lower frequency to complete a given task or to perform more tasks for a given frequency.
- Achieving the solution at lower cost, lower power than standard off-the-shelf processor cores

→ Code examples

DSP - Implementation in CodAL

```
inline s15_16_t cdsp_fp16_mult(s15_16_t x, s15_16_t y)
{
#ifdef L110_ACCELERATION
    s15_16_t z;
    asm volatile ("mul32fxps15_16 %0, %1, %2\t": "=&r"(z) : "r"(
    return z;

#else
    int64_t mulres = (int64_t)x*(int64_t)y;
    return cdsp_fp16_round(mulres);

#endif
}
```

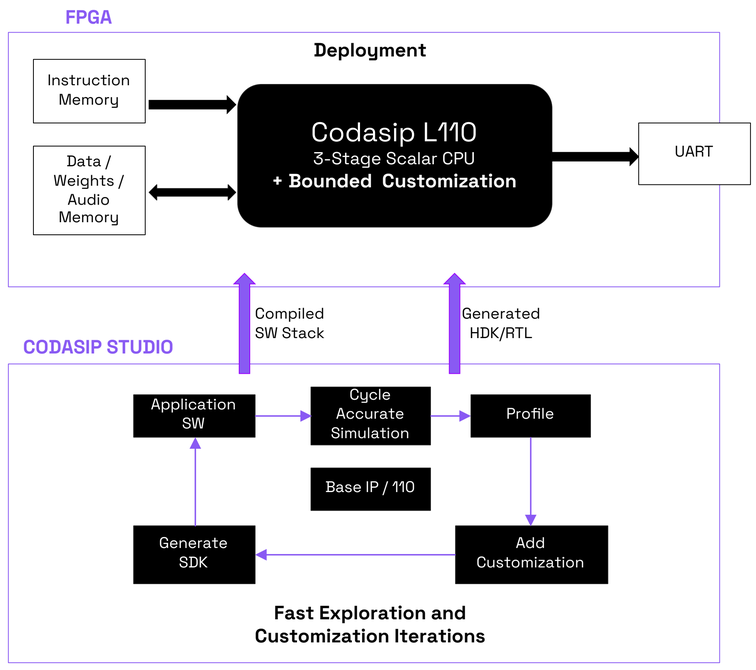
CodAL

```
stage pipe.ID
{
    src1 = rf_gpr.r0[rs1];
    src2 = rf_gpr.r1[rs2];
    src3 = rf_gpr.r2[rd];

    switch (opc) {
        default:
            mul32fxp_opc = (enum mul32fxp_op_e)MUL32FXP_NOP;
            break;
        case OPC_FXP_MUL32FXP_S15_16:
            mul32fxp_opc = (enum mul32fxp_op_e)MUL32FXP_MUL;
            break;
        ...
    }

    datatype = (enum fxp_datatype_e)opc[16..13];

}
```

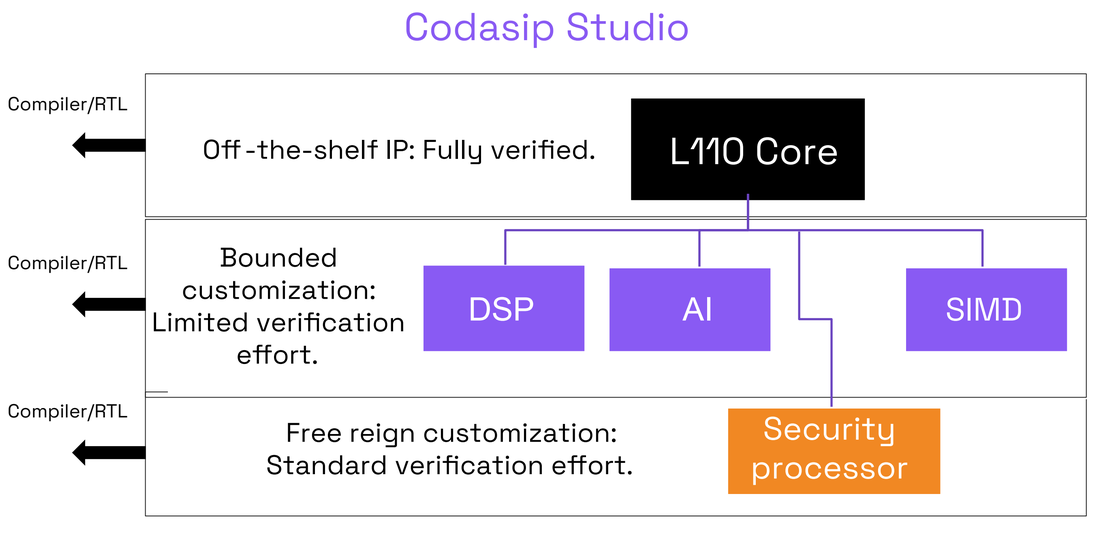
→ Emphasis on the SW/HW Co-Design paradigm

- Codasip IP + Bounded Customization Enables High Performance / Low Power tradeoff exploration and deployment.
- Emphasis on the SW/HW Co-Design paradigm, where you can optimize both SW and HW in one package.

Audio Anomaly Detection in Near-Sensor Embedded Devices
[System Level]



Audio Anomaly Detection in Near-Sensor Embedded Devices
[Design Flow Level]



→ Benefits of Custom Compute

Cost management

- One base core can spawn many application processors through custom compute (shared knowledge & tools)
- Integrated toolchain manages both software (SDK) and hardware development kits (HDK)
- Custom Bounded Instruction significantly reduces verification time and costs

Doing more with limited resources

- Bounded Customization enables Software Engineers to design processors
- Custom compilers through automation
- No harm verification tools minimizing verification effort

Increasing opportunity for profit

- Enter new markets through targeted acceleration
- Lower costs through just enough processing
- Optimal use of HW and SW engineering

Maximizing product's lifecycle and revenue through time to market

- Bounded Customization (BC) enables customization without detail processor knowledge
- Through BC, risk is minimized by limiting processor verification only to the new instructions
- Integrated HW & SW toolchain maintains consistency between HW & SW engineering