NetTimeLogic GMBH

ITSF 2024

CPUs vs MCUs vs FPGAs

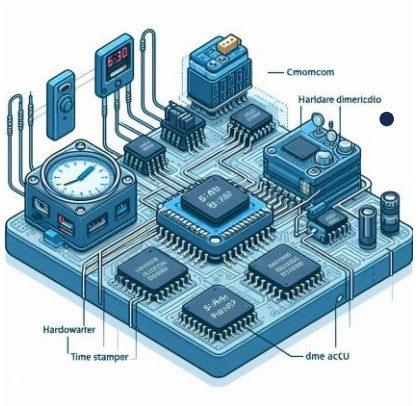
Timing Related Capabilities

Thomas Schaub, COO
thomas.schaub@nettimelogic.com

Sven Meier, CEO
sven.meier@nettimelogic.com

Overview

- **CPU**
  - Clock Speed: 0.5-5+ GHz
  - Memory and peripherals: Typically external
  - OS: Usually full scale like Windows, Linux etc.
- **MCU**
  - Clock Speed: 10-500 MHz
  - Memory and peripherals: Highly integrated often on-chip
  - OS: Baremetal or RTOS



- **FPGA**
  - Fmax: Low-End 50-250MHz, Mid-R up to 500MHz, High-End up to 1GHz
  - Memory and peripherals: BRAM integrated, peripherals integrateable
  - OS: No OS, only on SoCs CPUs or Soft-Core CPUs

*Graphic generated by AI

Timing Related Capabilities

Time Stamping

CPU

- TGPIO[1] for some CPUs with a matching Mainboard
- Often external Time Stamping Chip support
- Only limited number of timing capable IOs

MCU:

- Input Capture Units on MCUs
- Only limited number of timing capable IOs

FPGA:

- Functionality not out of the box
- Very high precision with TDC

[1]Timestamped General-Purpose Input/Output, an Intel feature used for precise event timing in real-time systems

Timing Related Capabilities

Signal Generation

CPU:

- Different kind of Timers that supports Signal Generation
- TGPIO for some CPUs with a matching Mainboard

MCU:

- Timers can often directly trigger outputs

FPGA:

- Functionality not out of the box
- Very high precision with DTC

Timing Related Capabilities

Frequency Generation

CPU:

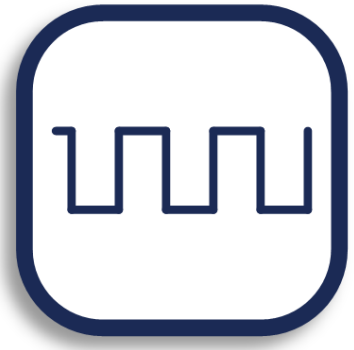
- Basically inexistent to generate frequencies on Pins
- TGPIO for some CPUs with a matching Mainboard

MCU:

- Typically kHz Frequencies based on timers

FPGA:

- Frequencies up to MHz ranges
- Functionality not out of the box
- Very high precision with DTC



Timing Related Capabilities

Time Synchronization/Distribution

CPU/MCU:

- **Often multiple clocks/time source in a system, distributed system**
  - Need to be kept in sync (cross-timestamping, PTM-like)
- **Firmware and driver support required**
  - TGPIO
  - PTM
  - E.g. phc2sys (PHC (PTP Hardware Clock) Framework)

FPGA:

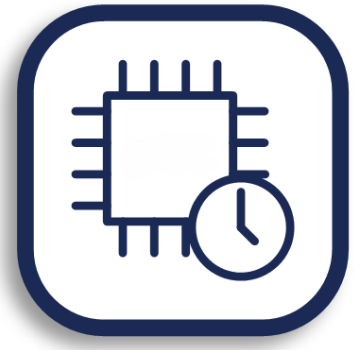
- **Time in the FPGA directly available, time can be exposed, frequency can directly be generated**
- **Only numerical**

Timing Related Capabilities

Time Adjustment

CPU/MCU:

- **Highly vendor dependent capabilities to steer frequency and phase of the system time**
  - Normally numeric adjustments
  - Cross timestamp based (e.g. like PTM)



FPGA:

- **Numeric adjustment of frequency and phase based on adding/subtracting additional time (ns, ps…) from the norm clock period**

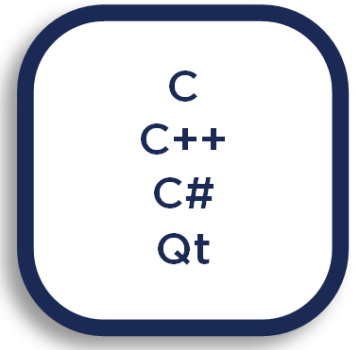
External Frequency adjustment directly on the Oscillator/PLL

Timing Related Capabilities
Software/Application

Final goal is to run applications synchronous!

CPU/MCU:

- **Timing aspects software/OS dependent**
  - RTOS
  - General Purpose OS like Linux
  - RT-Patches
  - Time triggers/ticks from the system clock

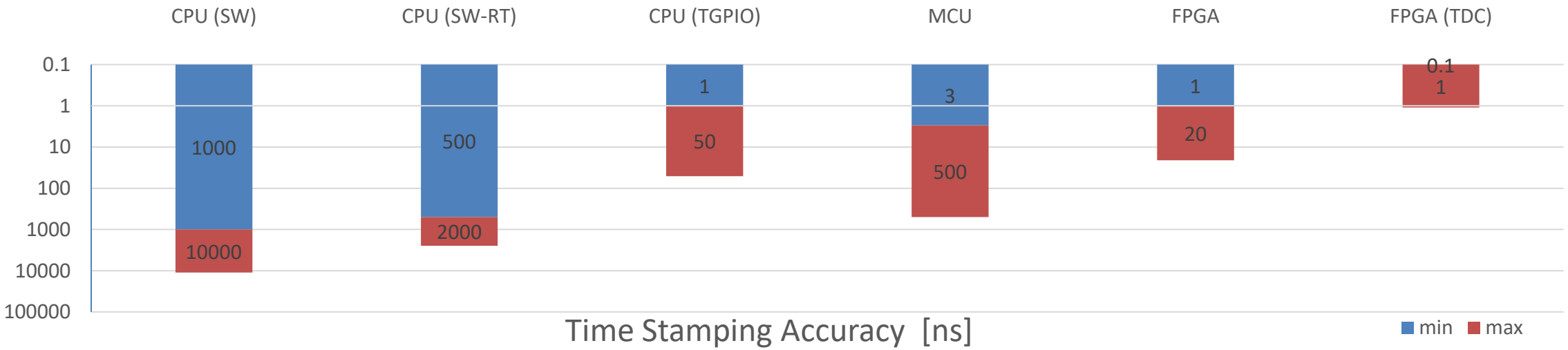


FPGA:

- **Trigger generation for software/app via IRQs or**
- **Direct application in the FPGA without IRQ**

Time Stamping

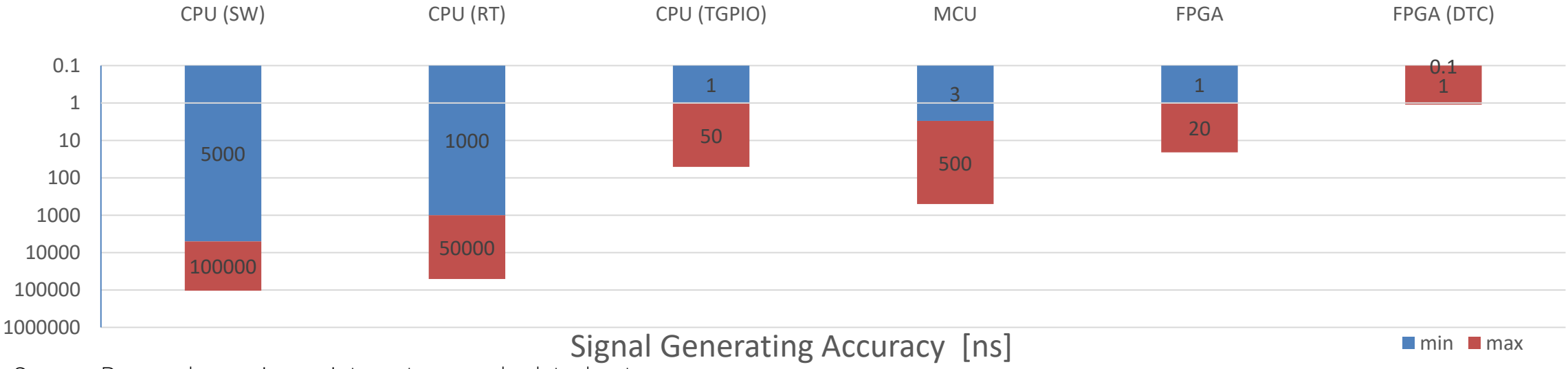
- **Time Stamping**
  - CPUs do not directly support Hardware Time Stamping (only TGPIO)
  - MCUs often have directly Hardware Time Stamping support (Input Capture Units)
  - FPGA Hardware Time Stamping can be implemented (also TDC)



Source: Personal experience, internet research, datasheets

Signal Generation
Frequency Generation

- **Accurate Signal/Frequency Generation**
  - CPUs have high precision timers to triggering events but mainly for software events
  - MCUs high resolution timers can directly control outputs timing
  - FPGA Hardware Trigger Generation can be implemented (also DTC)



Source: Personal experience, internet research, datasheets

Time Synchronization, Distribution and Adjustment

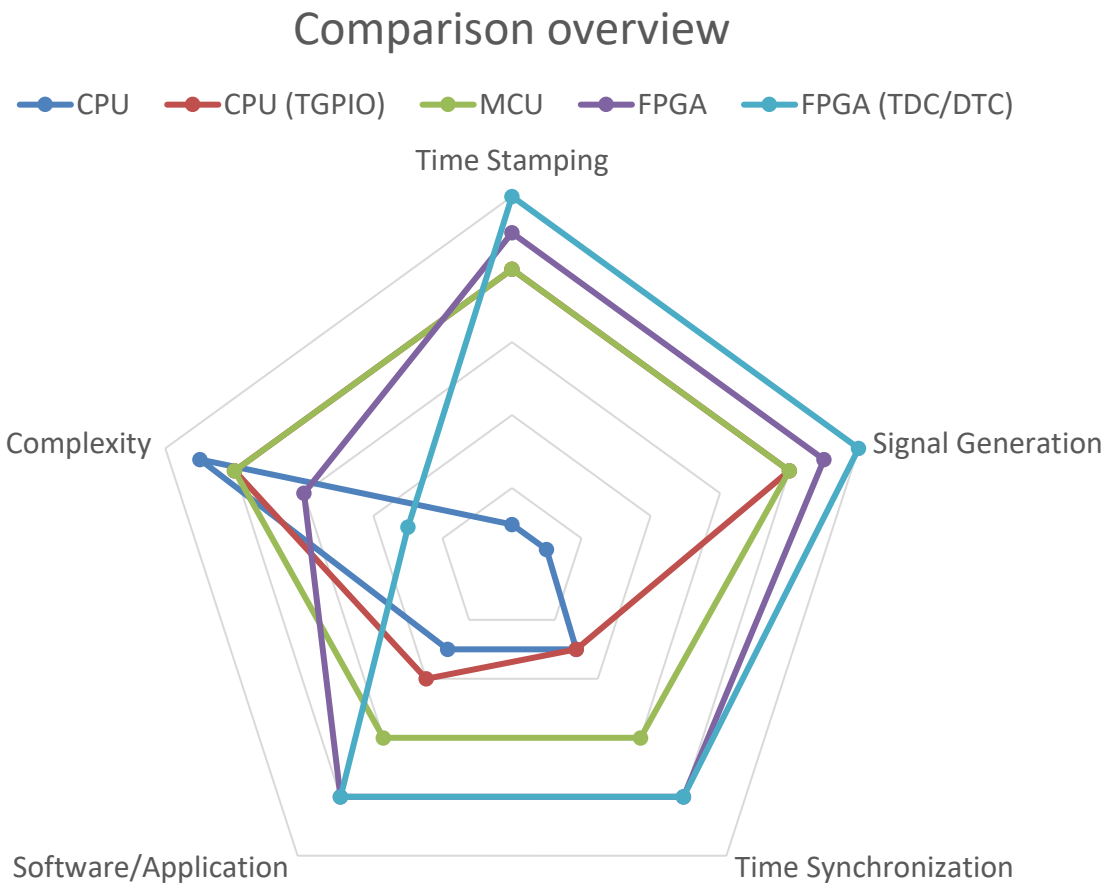
Based on Time Stamping, Signal/Frequency Generation, Software and IRQs.

- Time usually must be distributed (e.g. via PCIe, from an external PHC)
- Cross-Timestamping
  - To align frequency/phase of multiple clocks

Precise Synchronization does not help when the precise time can not be used!

Overview

Larger is better



Note: There are much more parameters, but they are not directly timing related.

Conclusion

- **Modern MCUs and CPUs support more and more timing related features**
  - TGPIO, Hardware Time Stamping, PTM, etc.
  - External chips help to provide the missing functionality and can have better performance than FPGAs
- **Once designed, MCUs and CPUs offer limited flexibility in hardware modifications**
- **Synchronized Time is more widely distributed in CPU Systems**
- **Often, integrating a CPU with an FPGA Co-Solution is advantageous**
- **For applications requiring high parallelism and multiple channels, FPGAs retain advantages, as each GPIO can achieve precise timing**

Conclusion

Ultimately, it is your application that will define the best fit for your needs

Thank you!

See Network Timing, Network Redundancy and our modular Hardware Platform AIONYX live at our booth!!!

www.nettimelogic.com

contact@nettimelogic.com