
Parallelism in hardware: IC design

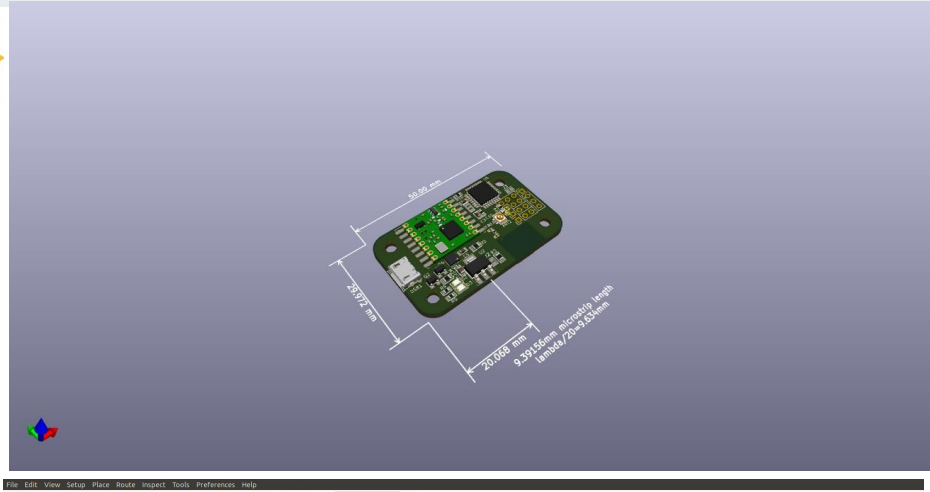
\IoT and embedded systems specialist

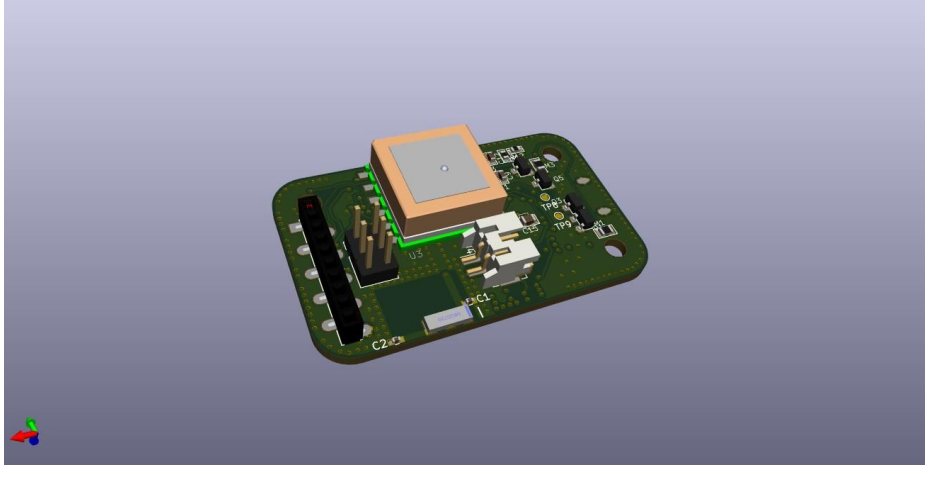
Alois Mbutura

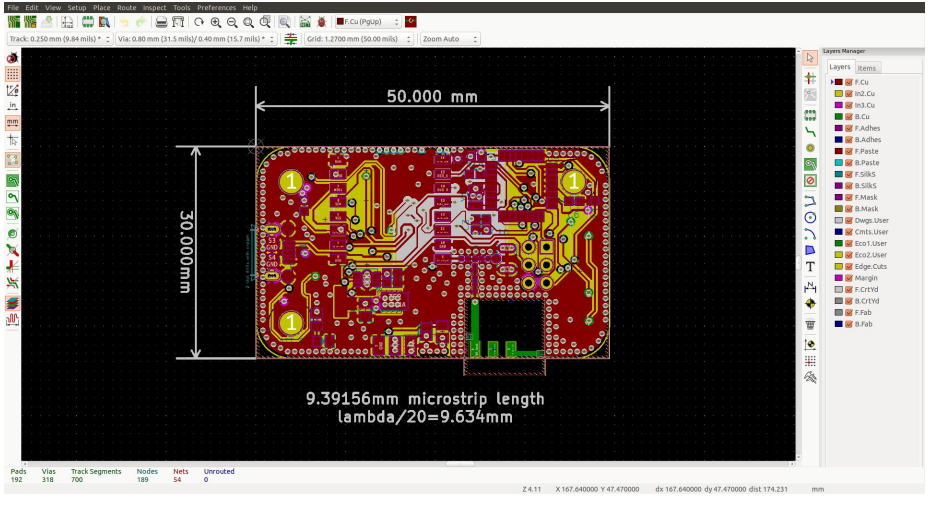
1. Started in 2012.
2. Currently work at Kaiote limited(kaiote.io) in Kenya as an IoT specialist.
3. Working on unified wireless, low power solutions with regard to water metering and water access.

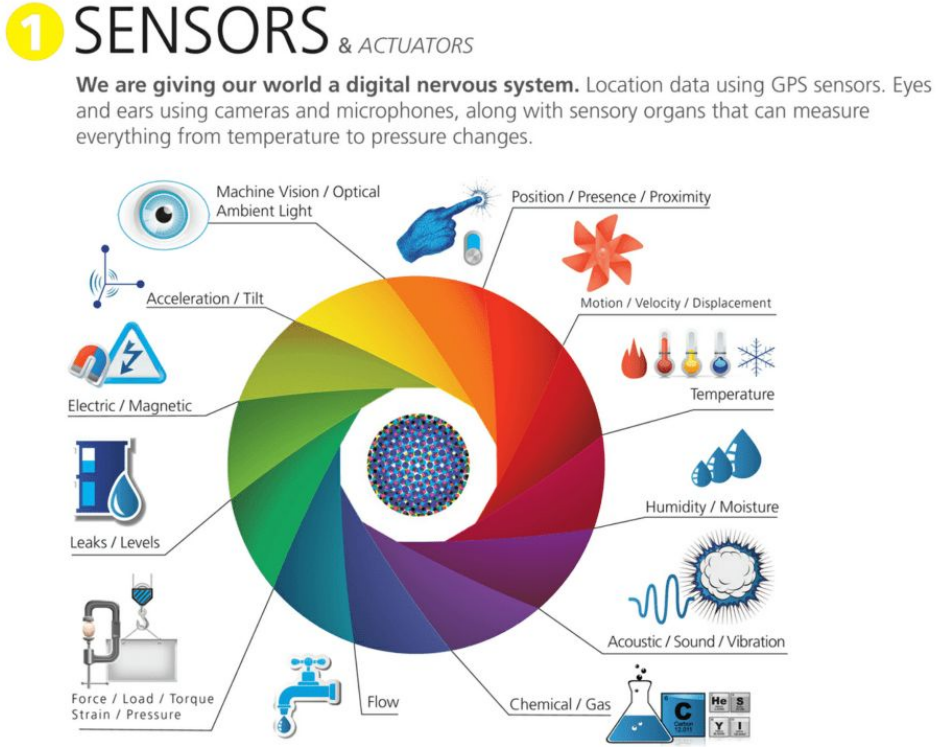
Sample embedded system

Product design and manufacture







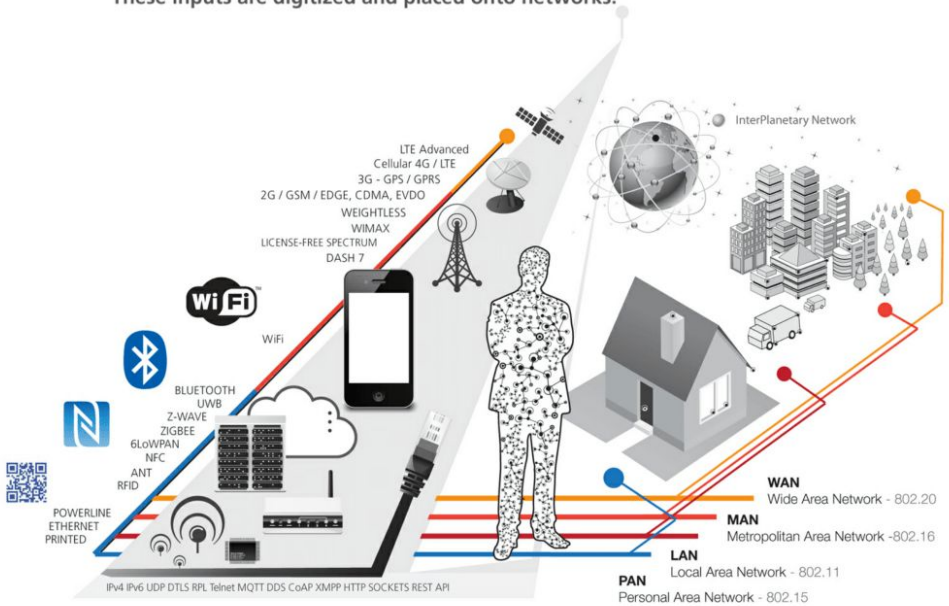


IoT

Involves the aggregation of data from physical sensors and its communication to another machines.

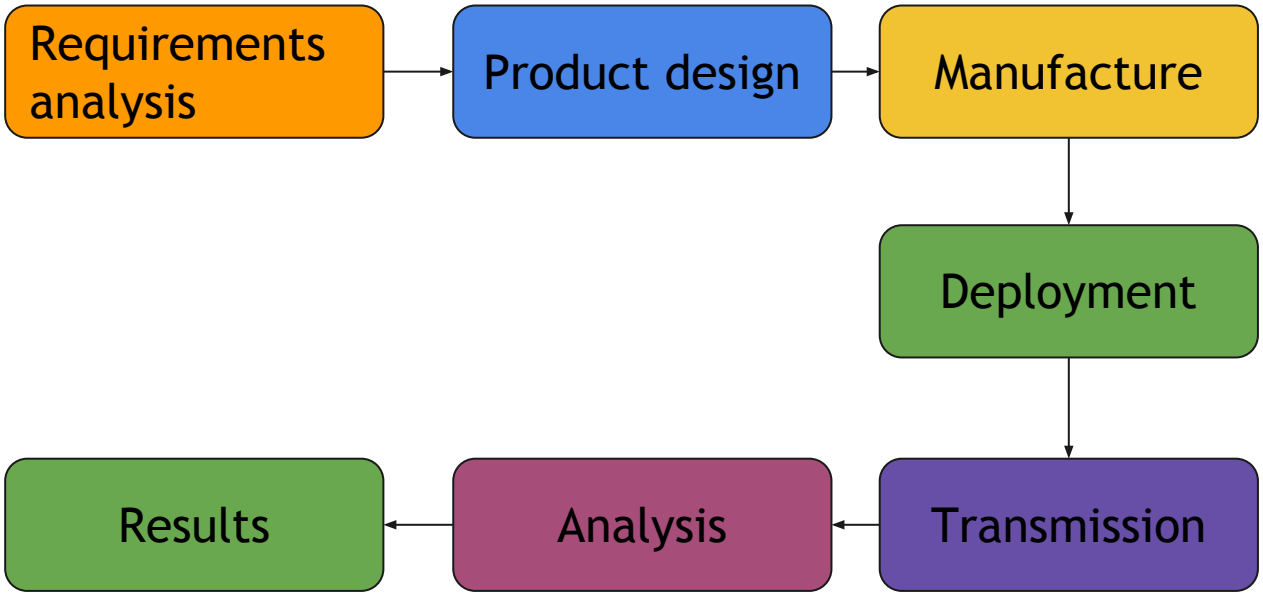
2 CONNECTIVITY

These inputs are digitized and placed onto networks.

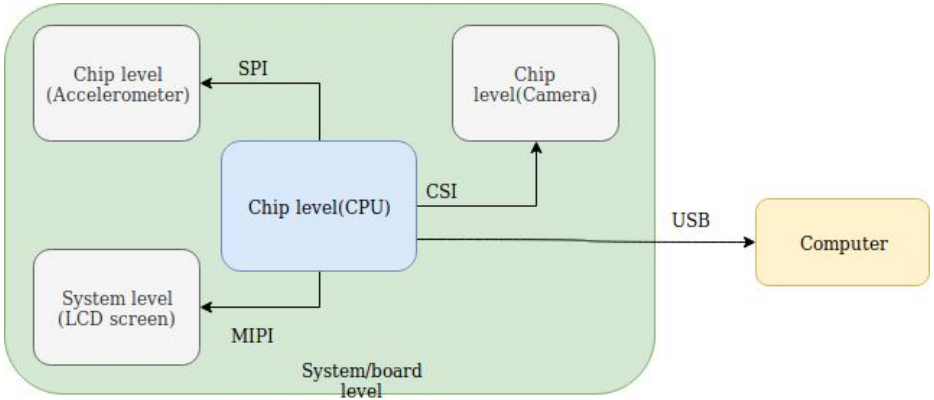


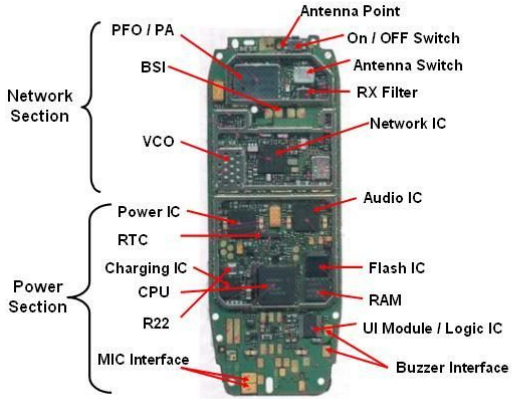
Due to the convenience of wireless
Communication, IoT is usually related to
Wireless but this is not true.

https://goo.gl/mio3rE


Requirements analysis
Product design
Manufacture
Deployment
Transmission
Analysis
Results

System level





NOTES:

1. **UEM =**
   Logic IC
   + Charging IC
   + Audio IC
   + Power IC
2. **PFO =**
   Antenna
   Switch
   + PFO
3. **Flash IC =**
   RAM + Flash
   IC

Embedded Computing platforms

Inflexible/hard computing platforms

- SOC(Systems on chip)
- GPU(Central processing unit)
- GPU(Graphical processing unit)
- Microcontrollers

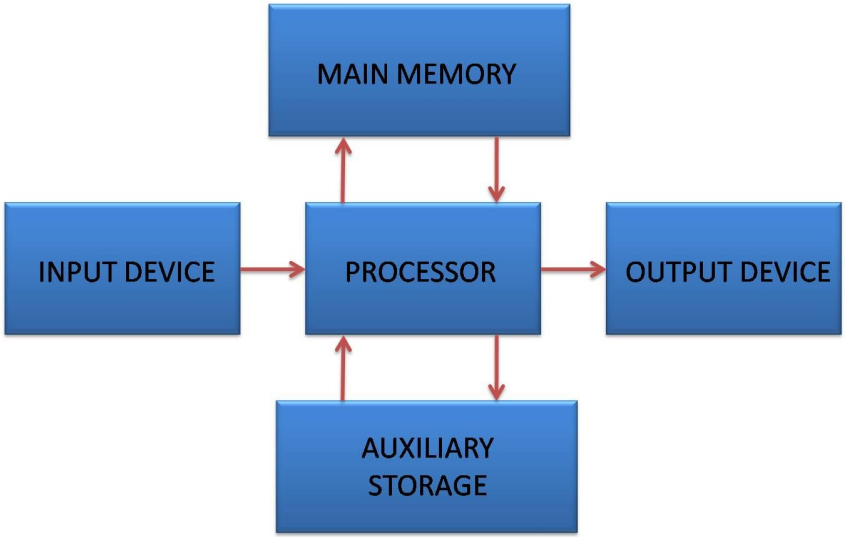
Flexible/soft computing platforms

- FPGA(Field programmable gate arrays)
- CPLD(Complex programmable logic devices)

Hybrid computing platforms

- APSOC(All programmable systems on chip)

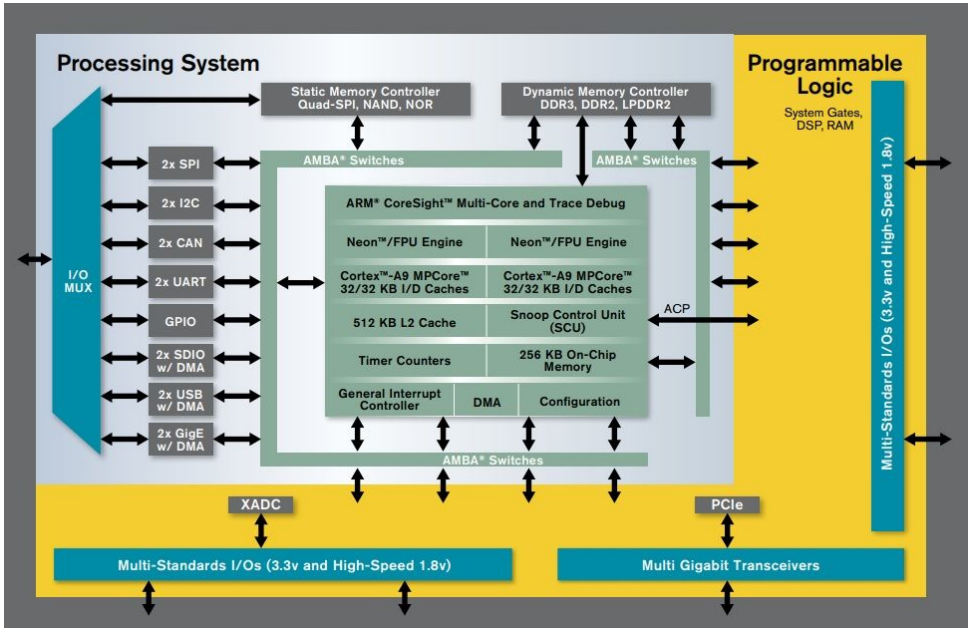
General computer system



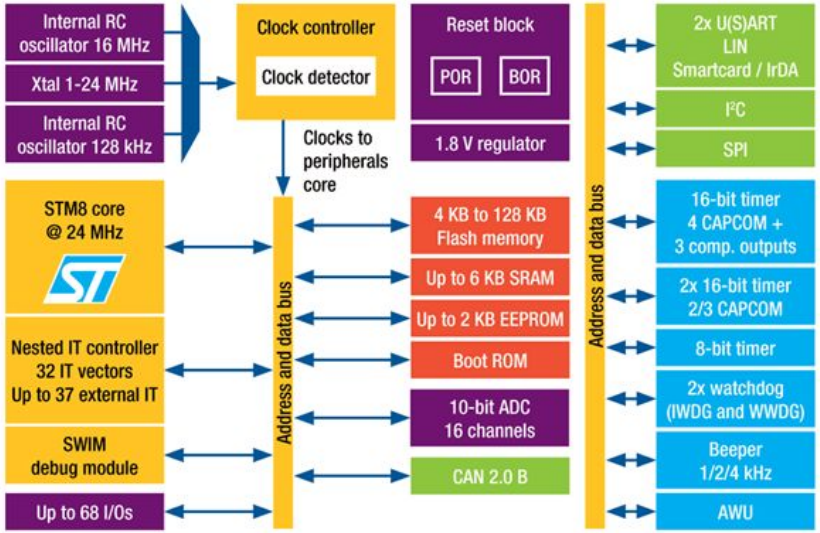
- **Memory Hierarchy**

(Main memory, cache, permanent storage)

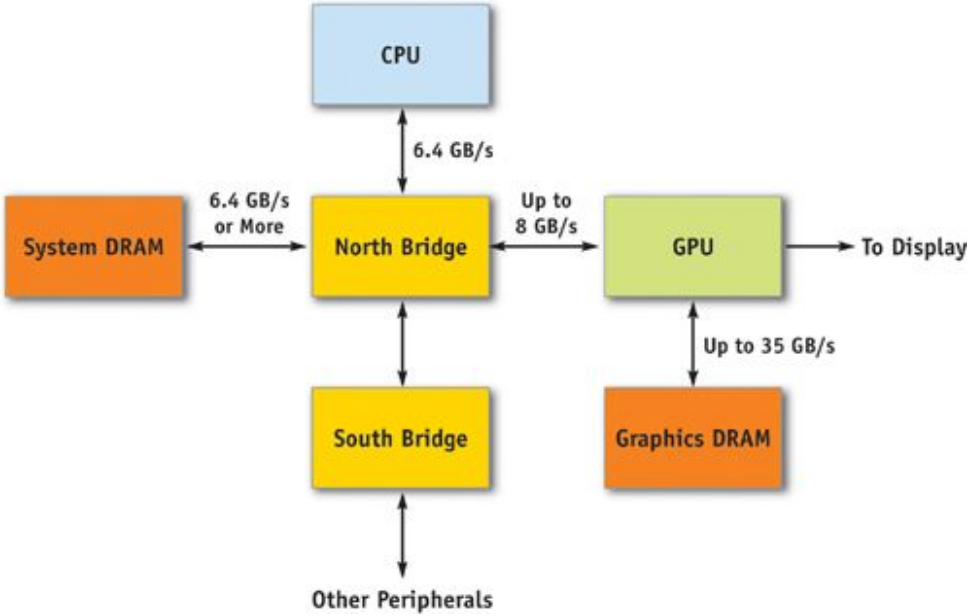
- **I/O(Input/output)**
- **Processing system**



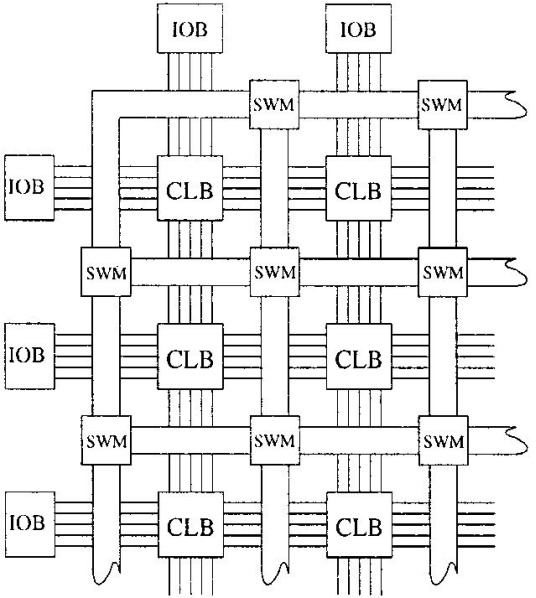
All programmable SOC 32 bit



Microcontroller 8 bit

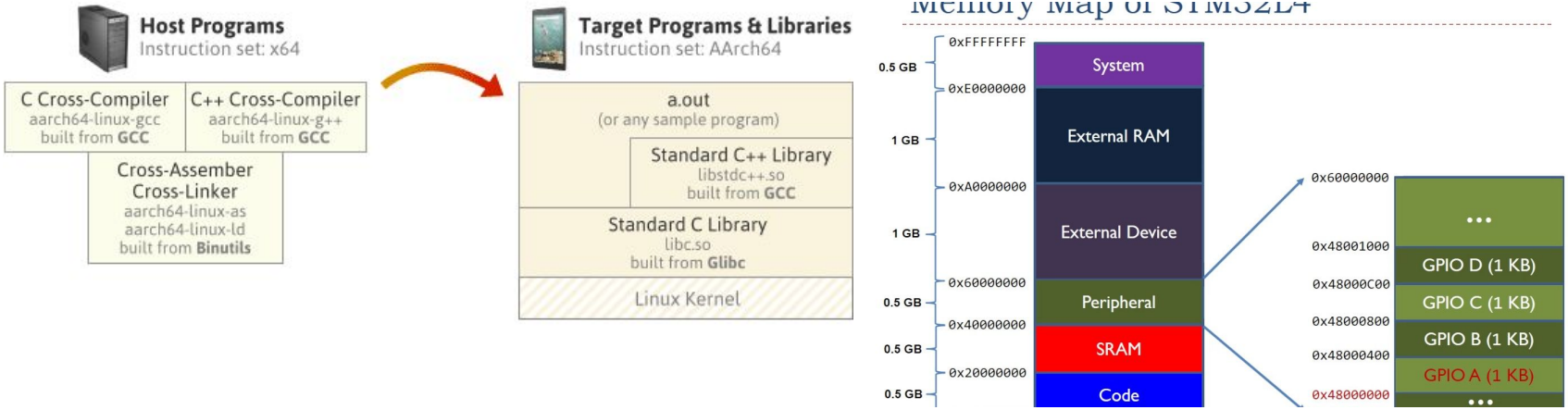


Heterogeneous x86(-64) system

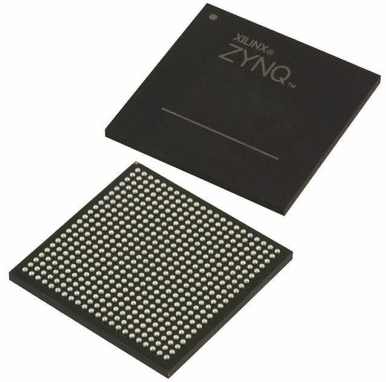


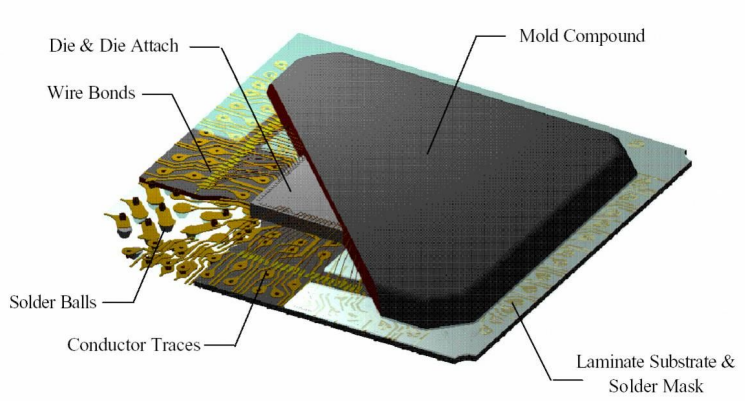
FPGA architecture

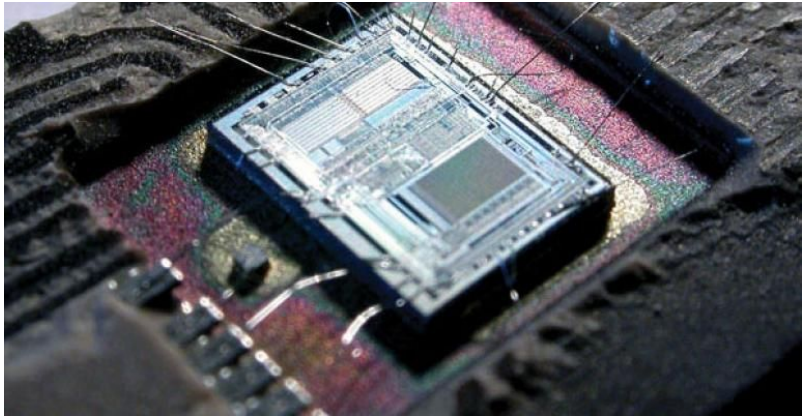
Developmental flow

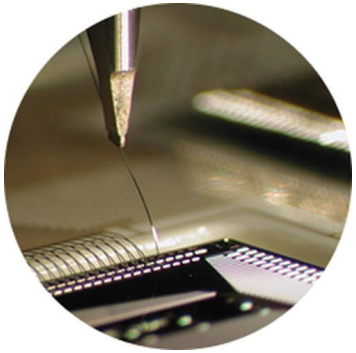


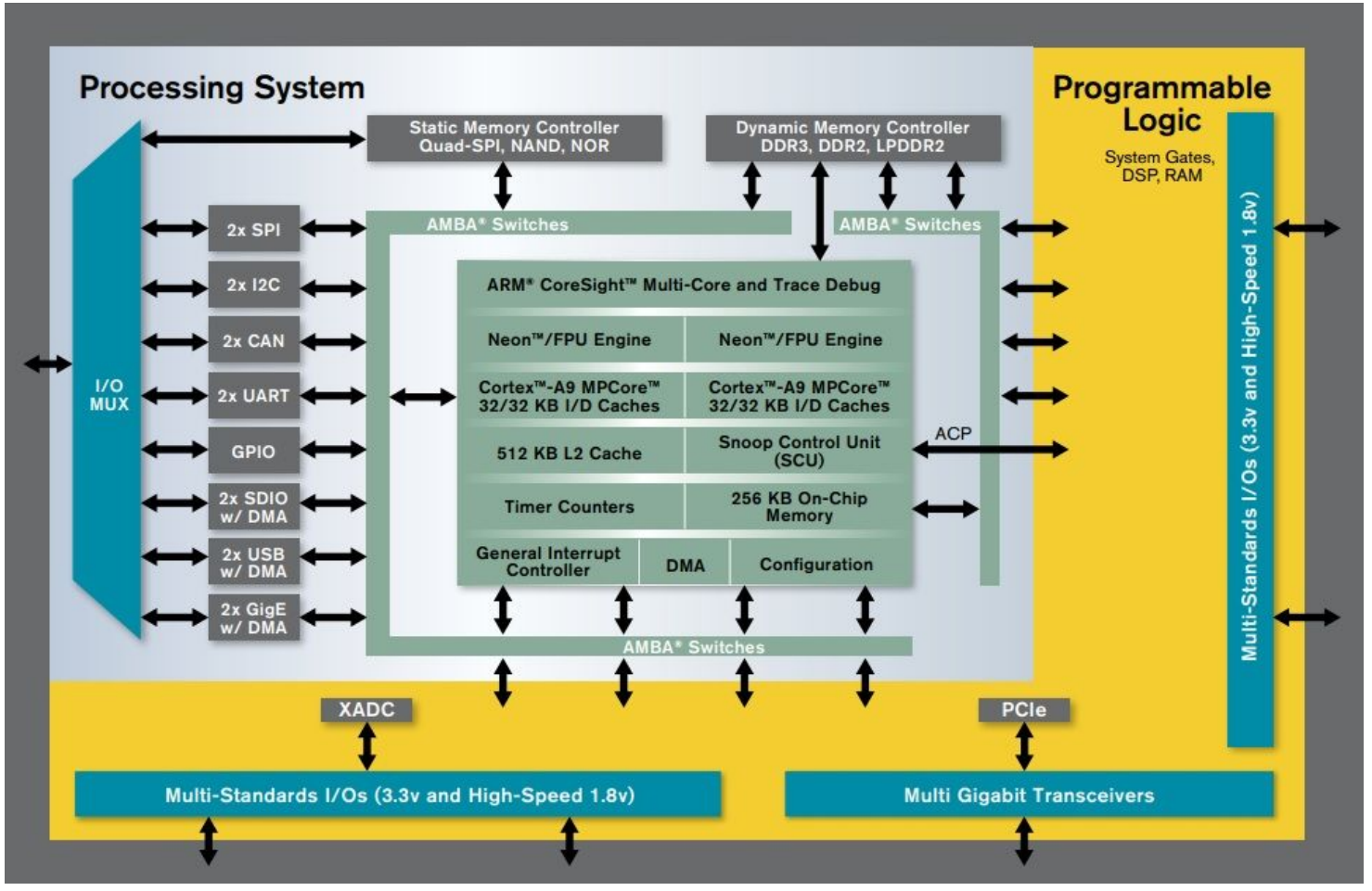
Chip level


XILINX
ZYNQ

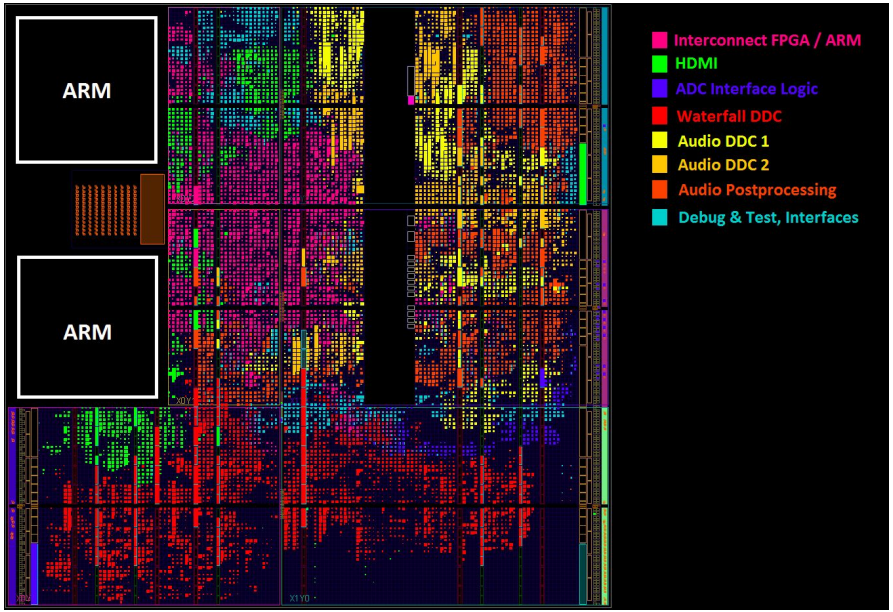

Die & Die Attach
Wire Bonds
Mold Compound
Solder Balls
Conductor Traces
Laminate Substrate &
Solder Mask



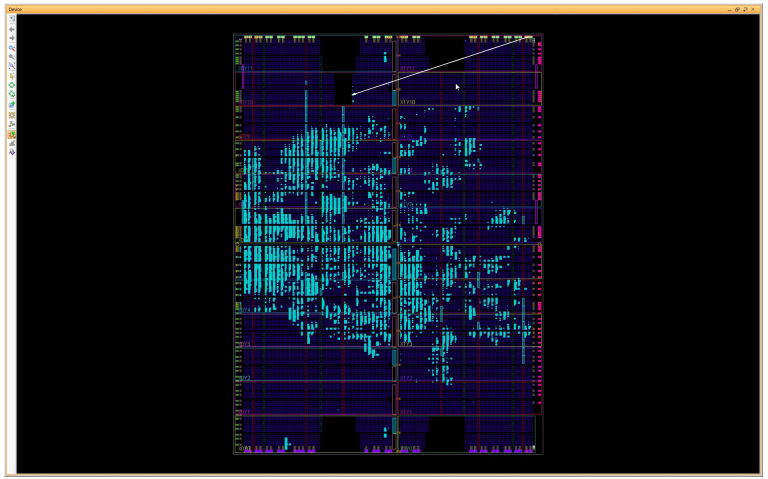



Processing System
Programmable Logic
System Gates, DSP, RAM
Static Memory Controller Quad-SPI, NAND, NOR
Dynamic Memory Controller DDR3, DDR2, LPDDR2
I/O MUX
2x SPI
2x I2C
2x CAN
2x UART
GPIO
2x SDIO w/ DMA
2x USB w/ DMA
2x GigE w/ DMA
AMBA® Switches
AMBA® Switches
ARM® CoreSight™ Multi-Core and Trace Debug
Neon™/FPU Engine
Neon™/FPU Engine
Cortex™-A9 MPCore™ 32/32 KB I/D Caches
Cortex™-A9 MPCore™ 32/32 KB I/D Caches
512 KB L2 Cache
Snoop Control Unit (SCU)
ACP
Timer Counters
256 KB On-Chip Memory
General Interrupt Controller
DMA
Configuration
AMBA® Switches
Multi-Standards I/Os (3.3v and High-Speed 1.8v)
XADC
PCIe
Multi-Standards I/Os (3.3v and High-Speed 1.8v)
Multi Gigabit Transceivers

APSOC and FPGA internal view



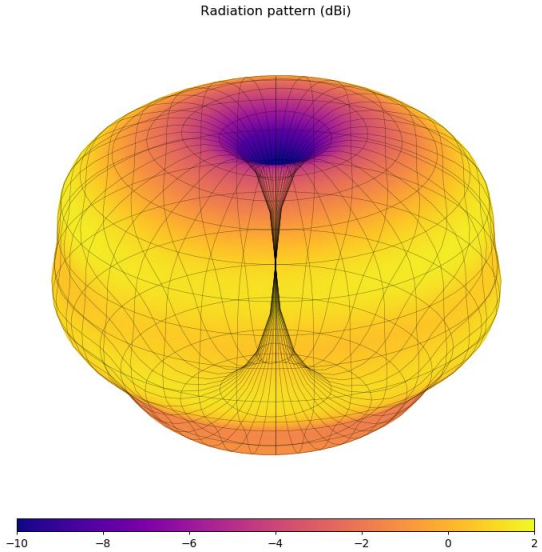
APSOC

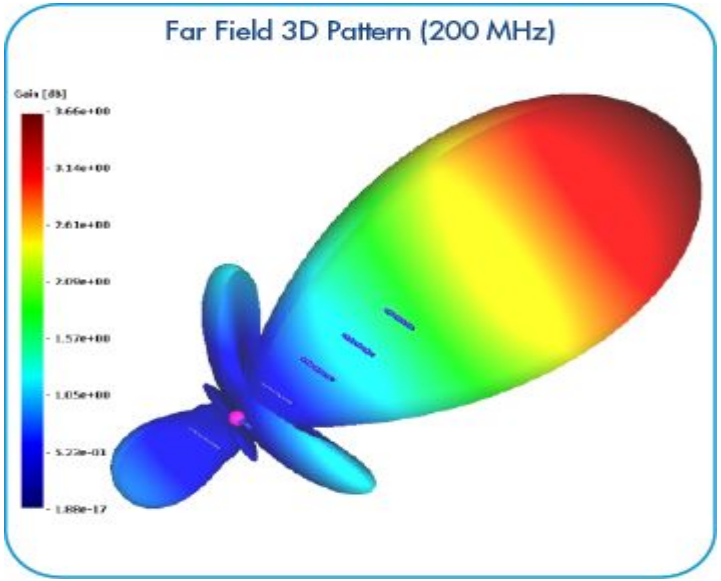


FPGA

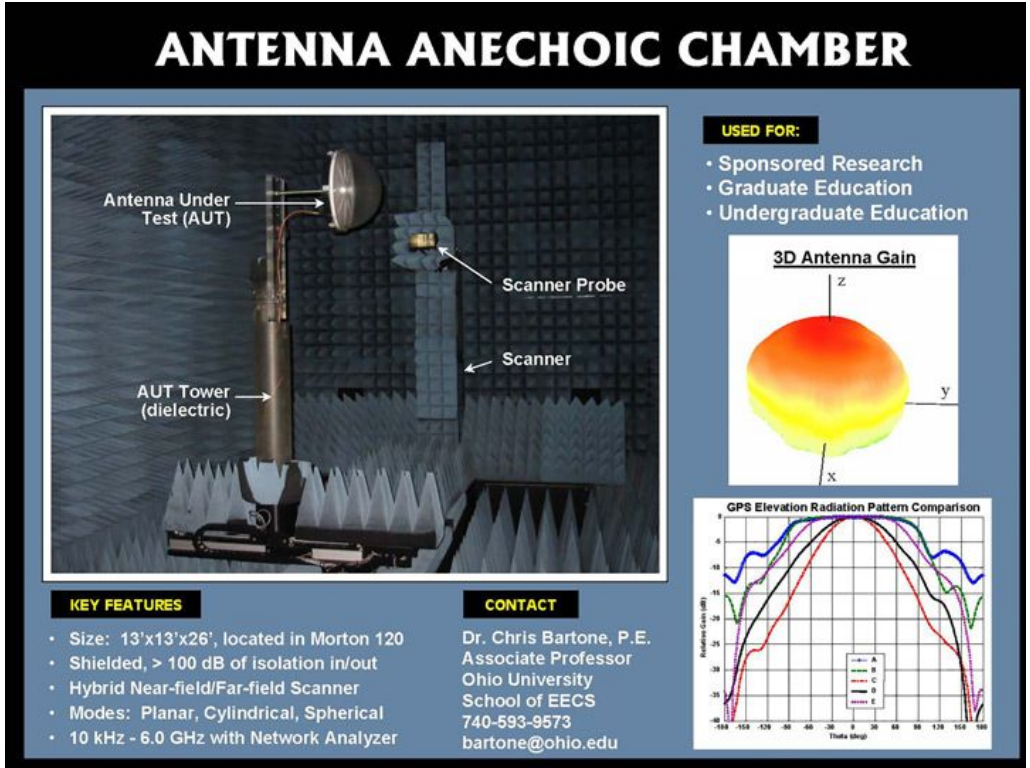
Product design

Sample design:Antenna field visualisation by simulation





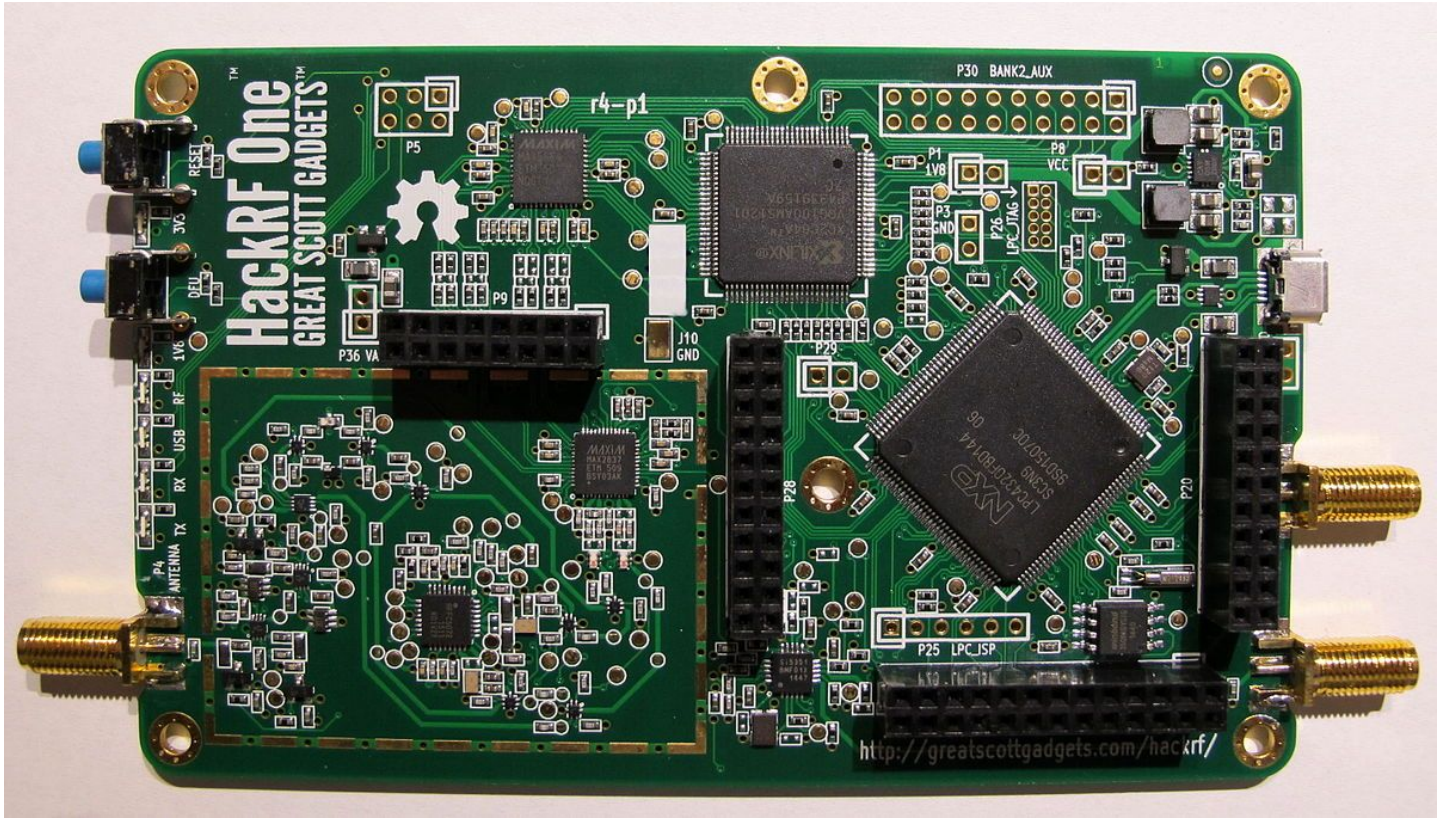
Sample design:Antenna field visualisation by testing(Anechoic chamber)



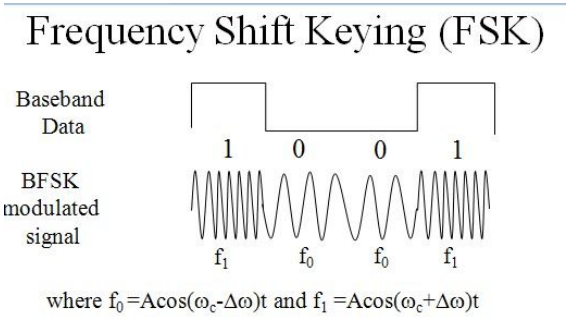
Benefits of hardware parallelism in FPGA and APSOCs

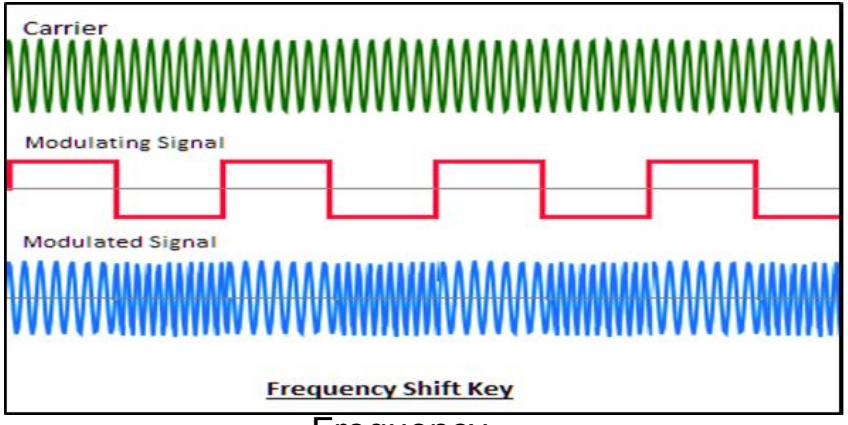
1. Custom protocols(Wired and wireless)
2. Interfacing to older physical systems(Glue logic)
3. Implementing flexible designs
4. Custom codec designs
5. Computer vision
6. Signal processing

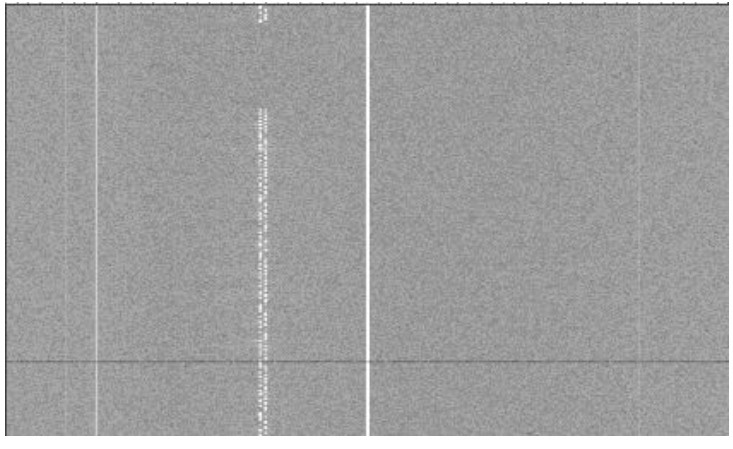
CPLD and FPGA on software defined radio

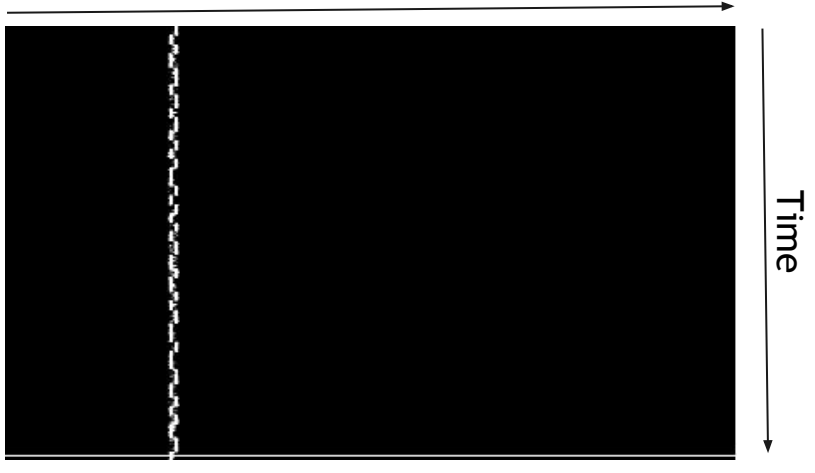


Car remote Keyfob(open/close)

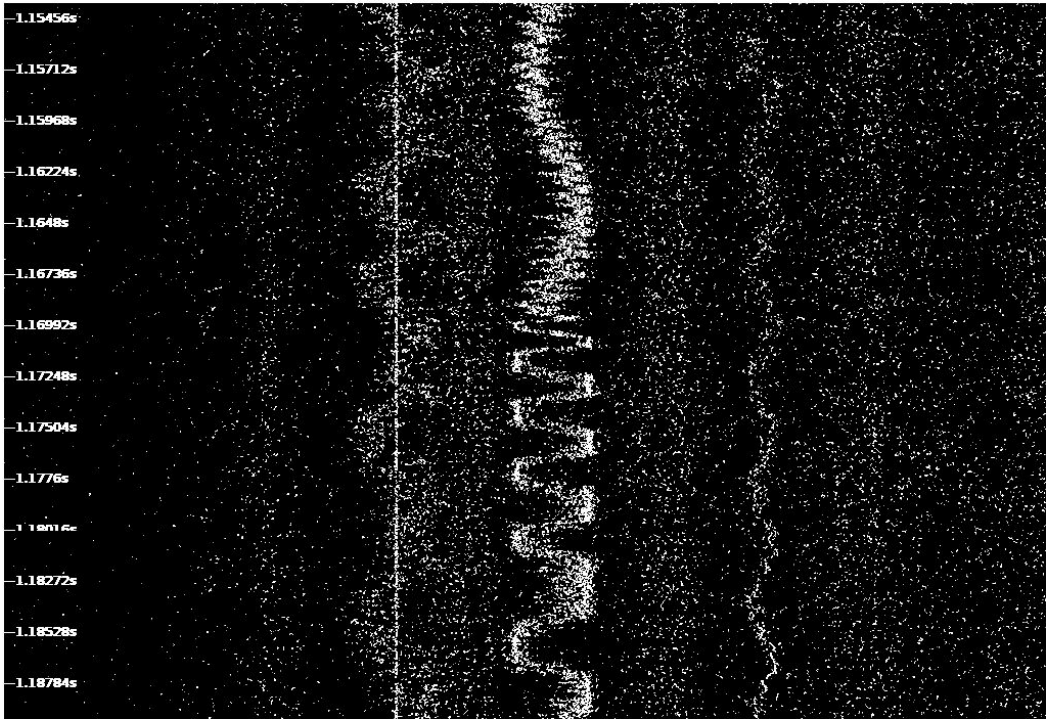


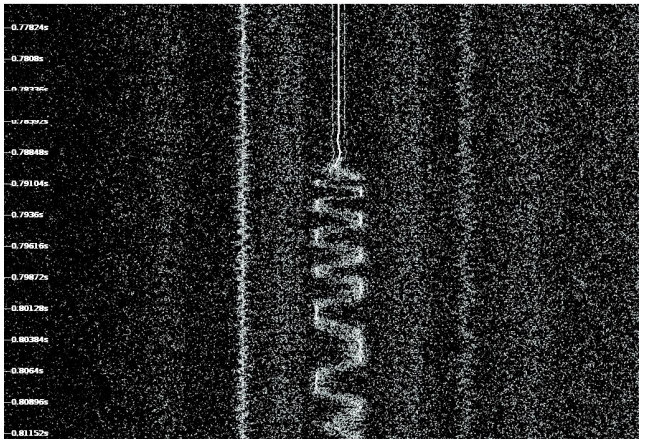


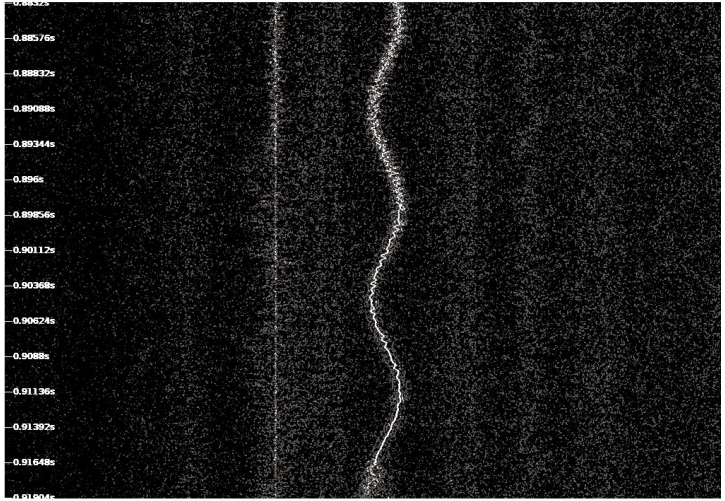




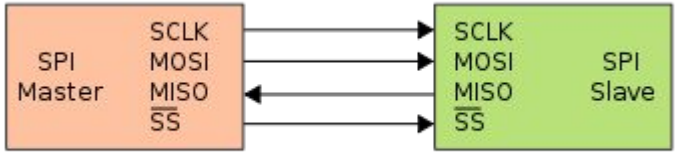
Analog FM(Frequency Modulation) on a local FM radio station talk show on a weekday night







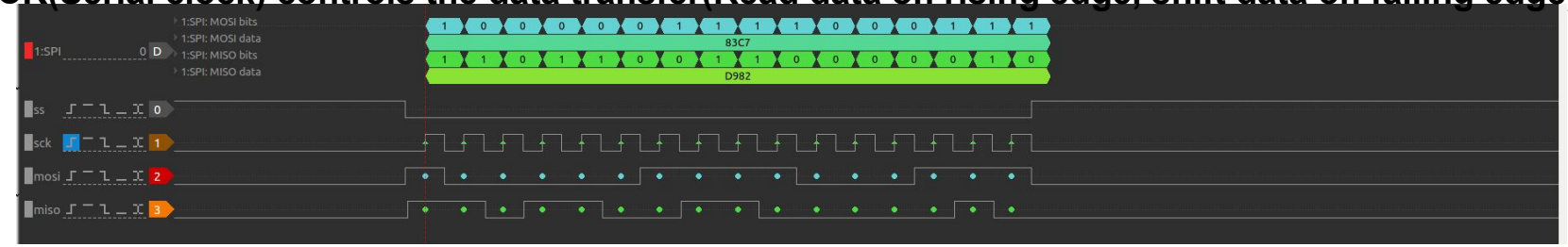
Demo design: microcontroller interfacing to FPGA over Serial Peripheral Interface(SPI)



Physical layer: 4 wires for comms

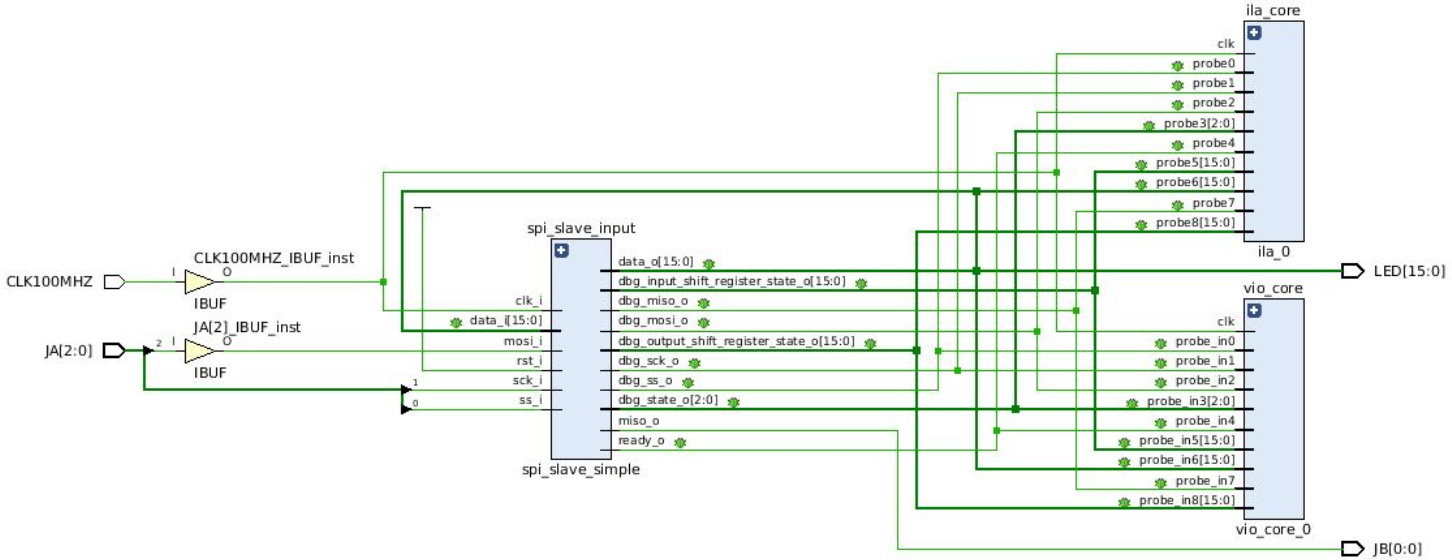
Demo full 2 byte duplex read while write over SPI

SCK(Serial clock) controls the data transfer(Read data on rising edge, shift data on falling edge)



Block diagram of SPI after FPGA synthesis

SPI on left, debugging signal monitors on the right(ila, vio)



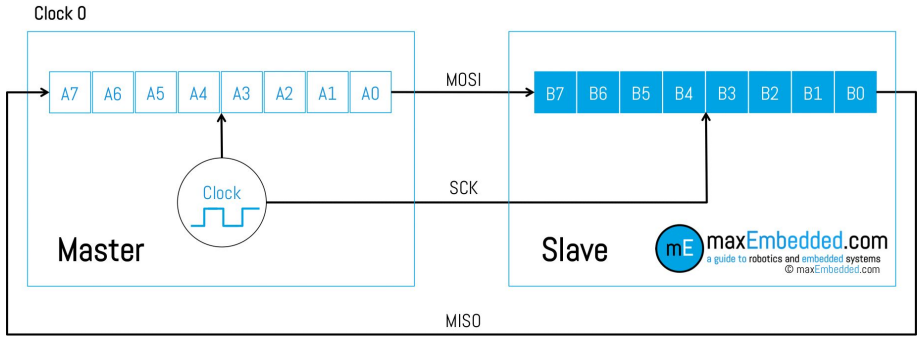
```
I (11886) [FPGA_SPI_SLAVE]: Ledstate value is 0x7589
I (11886) [FPGA_SPI_SLAVE]: Note: SPI read success
I (11887) [FPGA_SPI_SLAVE]: Value read is 0x5e10
I (11892) [FPGA_SPI_SLAVE]: Value match :)

I (11907) [FPGA_SPI_SLAVE]: Ledstate value is 0x0644
I (11907) [FPGA_SPI_SLAVE]: Note: SPI read success
I (11908) [FPGA_SPI_SLAVE]: Value read is 0x7589
I (11914) [FPGA_SPI_SLAVE]: Value match :)

I (11929) [FPGA_SPI_SLAVE]: Ledstate value is 0xa996
I (11929) [FPGA_SPI_SLAVE]: Note: SPI read success
I (11930) [FPGA_SPI_SLAVE]: Value read is 0x0644
I (11935) [FPGA_SPI_SLAVE]: Value match :)

I (11950) [FPGA_SPI_SLAVE]: Ledstate value is 0xa70e
I (11950) [FPGA_SPI_SLAVE]: Note: SPI read success
I (11951) [FPGA_SPI_SLAVE]: Value read is 0xa996
I (11956) [FPGA_SPI_SLAVE]: Value match :)

I (11971) [FPGA_SPI_SLAVE]: Ledstate value is 0xd74d
I (11971) [FPGA_SPI_SLAVE]: Note: SPI read success
I (11973) [FPGA_SPI_SLAVE]: Value read is 0xa70e
I (11978) [FPGA_SPI_SLAVE]: Value match :)
```




Q&A