Free, opensource Field Programmable Gate Array (FPGA) development frameworks for radiofrequency communication – Digital communication using GNU Radio

N. Gallone, G. Goavec-Merou, J.-M Friedt

FEMTO-ST/time & frequency department

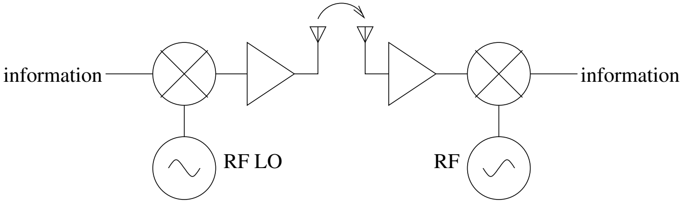
jmfriedt@femto-st.fr

slides and references at `https://github.com/oscimp/amaranth_twstft`

June 11, 2022

Architecture of an SDR transmitter

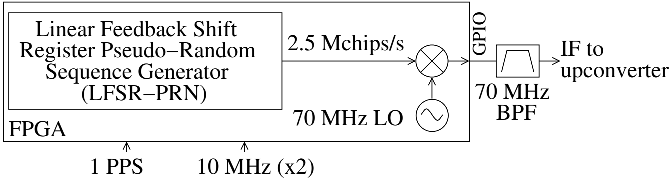
- FPGA (Field Programmable Gate Array)-only SDR transmitter, no external analog component other than a band-pass filter and antenna
- Baseband information is frequency-transposed by mixing with a local oscillator (LO) to a radiofrequency band
- IQ modulator requires generating a complex ($\exp(j\omega t)$) LO
- LO is generated within the FPGA at a frequency limited to a fraction of the FPGA internal-clock frequency
- square output on a FPGA General Purpose Intput Output (GPIO) pin allows for using harmonics if needed



Outline

How to generate a modulated radiofrequency signal on an FPGA configured as SDR transmitter?
Amaranth (formerly nMignen) is selected for its

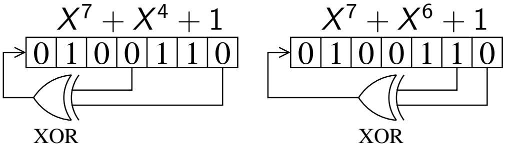
- flexibility (**Python-like** description of the hardware)
- **independence** from vendor-specific tools (Vivado framework, Mathworks HLS tools)
- **fast synthesis** for fast development cycle (program-simulate-test)
- compatibility with a **wide range** of lower-end FPGAs [1]
- practical demonstration on the Zynq7020 fitted on the Zedboard[2]



[1] https://github.com/amaranth-lang/amaranth
[2] https://www.avnet.com/wps/portal/us/products/avnet-boards/avnet-board-families/zedboard/

Baseband signal structure

- We select a pseudo-random sequence generated by a Linear Feedback Shift Register (LFSR) for assessing Bit Error Rate (BER) and used for ranging measurements
- Accurate time of flight measured by correlating a local copy of the transmitter LFSR sequence
- Modulation scheme: start with Binary Phase Shift Keying (BPSK), and later consider Quad-Phase Shift Keying (QPSK)



Getting started with FPGA basics and Amaranth programming

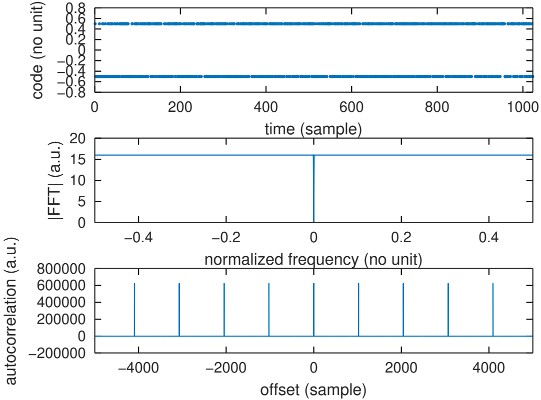
Baseband signal generation using Amaranth

1. DONNER LES PRINCIPES D'AMARANTH, COMBINATOIRE v.s SYNCHRONE
2. SYNTAXE D'AMARANTH
3. COMMENT PROGRAMMER LE LFSR ET SIMULER LA SORTIE DANS GTKWAVE

Baseband signal characterization ("random" = no repeating pattern)

Output sequence dumped as a binary file (here 10-bit long sequence)

```
def write_prn_seq(file, bitlen, code, seed=1, mode=1, seqlen = 2500000):
  with open(file,"wb") as f:
    v = seed
    for i in range(seqlen):
      f.write((v%2).to_bytes(1,byteorder='big'))
      v = nextstate(v,code,bitlen)
    f.close()
```



is read from Matlab or GNU/Octave with `f=fopen('file.bin');c=fread(f,inf,'int8');`

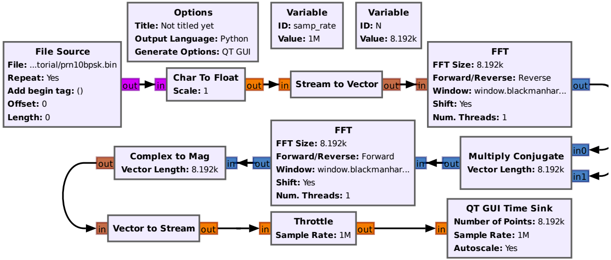
Correlation: remove mean value `c=c-mean(c);` Signal Processing toolbox `xcorr(code);` or `fcode=fft(code);iff(fcode.*conj(fcode));`

- autocorrelation and power spectrum density ...
- ... using the simulation output or an FFT analyzer

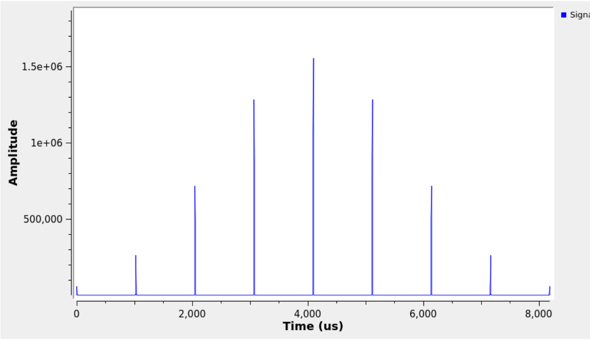
Baseband signal characterization using GNU Radio

Based on the convolution theorem $FFT(conv(x,y)) = FFT(x)\cdot FFT(y)$ and the close relation between convolution $conv(x,y)(\tau) = \int x(t)y(\tau-t)dt$ and correlation $xcorr(x,y)(\tau) = \int x(t)y(\tau+t)dt$ then

$$FFT(xcorr(x,y)(\tau)) = FFT(t)\cdot FFT^*(y)$$



Flowchart

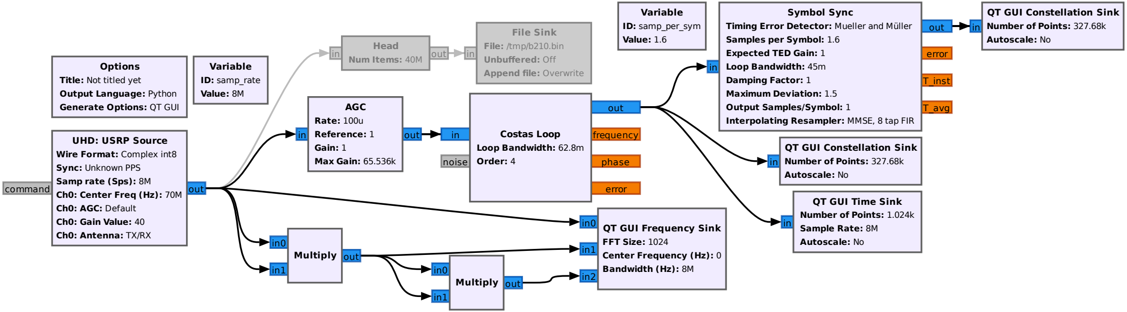


Result (correlation peak every 1023 sample delays)

File Source reads the file generated by Amaranth (1 byte/PRN chip – use Packed to Unpacked if 1 bit/PRN chip)

Transposition to radiofrequency band: BPSK

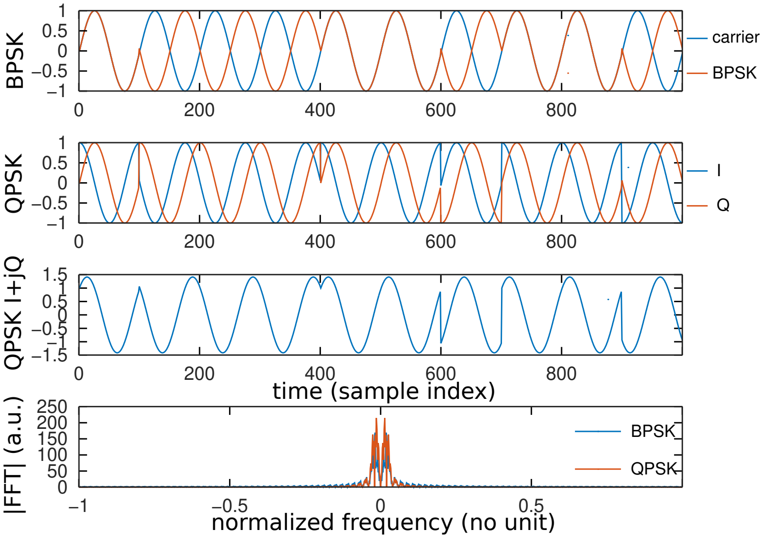
- Generating the Numerically Controlled Oscillator (NCO) as LO, selecting the clock source
- Mixing the baseband signal to LO to generate the RF signal: BPSK, $\varphi = \{0, \pi\} \Rightarrow s = \{-1, 1\} \in \mathbb{R}$
- Receiving and characterizing the RF signal (auto-correlation of the signal collected by the B210)
  - N-th power of N-PSK recovers the carrier
  - Costas Loop to cancel FPGA clock to B210 clock offset
  - Displaying the constellation, p samples/symbol including transitions
  - Symbol Synchronization block and displaying the constellation, 1 sample/symbol



Correlation with the local copy of the code: interpolate the code sequence to match the sampling rate: `code=repelems(code,[[1:length(code)] ; ones(1,length(code))*2]);`

Transposition to radiofrequency band: QPSK

- BPSK only uses half the bandwidth by modulating I and keeping Q null
- Double digital bandwidth at the same analog bandwidth: QPSK
- $\varphi = \{0, \pi/2, \pi, 3\pi/2\} \Rightarrow s = \{j, -1, -j, 1\}$ requires full complex modulator ...
- ... and LO$< f_{FPGA}/4$
- Practical implementation using Amaranth
- Receiving and characterizing the RF signal (auto-correlation of the signal collected by the B210)



Transposition to radiofrequency band: Frequency Modulation (FM)

- FM modulation: incoherent (no need to recover the carrier) modulation scheme still popular with ham radio (NBFM) and commercial broadcast (WBFM)
- $s(t) = \cos(2\pi f_c t + f_\Delta \int x(t)dt)$ with f_Δ the frequency deviation induced by signal x to be transmitted
- if $x(t) = \cos(2\pi f_m t)$ (one spectral component of signal x) then $s(t) = \cos(2\pi f_c t + \frac{f_\Delta}{f_m}\sin(2\pi f_m t))$
- $\Rightarrow$ tune f_c with $\sin(2\pi f_m t)$ to transmit the FM-modulated tone f_m
- $f_\Delta / f_m \ll 1$: NBFM, $f_\Delta / f_m \gg 1$: WBFM
- Practical implementation using Amaranth: generate a sine wave at f_m and update the LO frequency with this signal

Conclusion

PEUT-ON CONCLURE EN SYNTHETISANT LE MEME PROJET SUR UNE AUTRE BOARD ?

Supporting resources available at `https://github.com/oscimp/amaranth_twstft`