

# Digital Synthesis SP

Lab Sessions (v1.0)

Ing. Lukas Rondelez  
KU Leuven  
2017-2018



## 0 Credits

- ▶ Based on the original course of Van Landeghem D.
- ▶ Edits of my predecessors:
  - Meel J.
  - Goedemé T.
  - Van Beeck K.

LABORATORIUM  
DIGITALE  
SYNTHÈSE



Figure: original course material

# 0 Outline

① Introduction

② DSSS

③ Assignment

# 1 Outline

① Introduction

② DSSS

③ Assignment

# 1 B-KUL-YI6810: Digital Synthesis SP: Lab Sessions

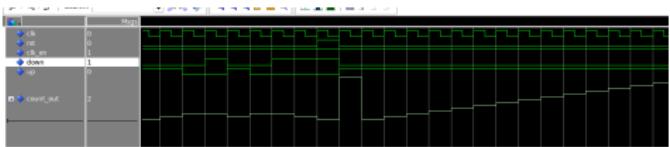
- ▶ 12 2-hour lab sessions
- ▶ Permanent evaluation + last lab: evaluation session
  - Digitale Synthese SP:
    - Digitale Ontwerpmethodologie: 20%
    - Digitale Synthese (theorie): 40%
    - **Digitale Synthese (practica): 40%**
- ▶ Second chance:
  - Complete/Correct the project (at home)
  - Come and show a demo (cfr. Last session)

# 1 The Project

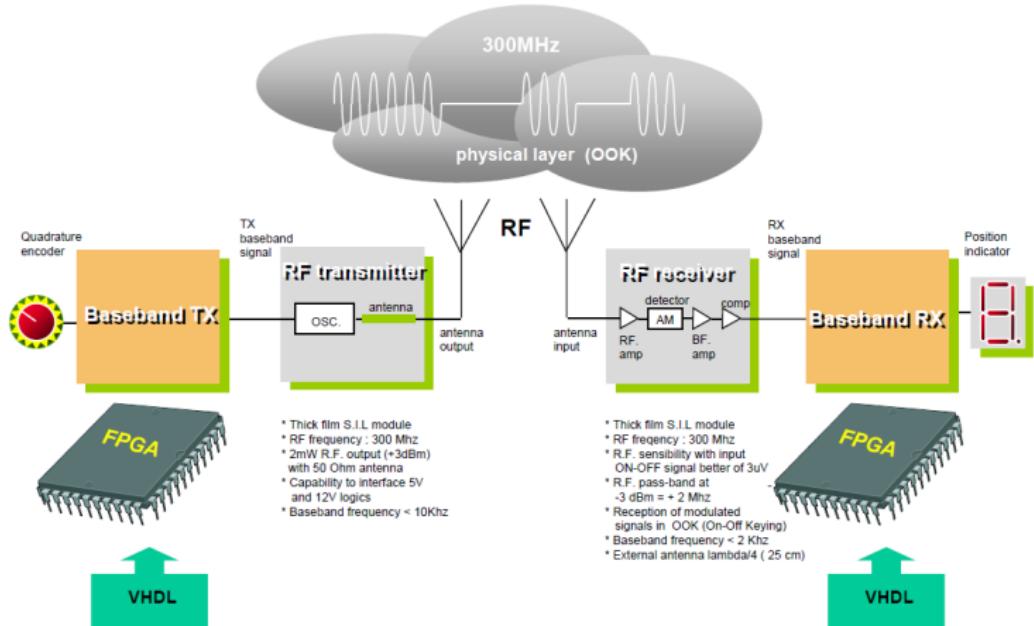
## ► What?

- Digital Electronics: VHDL  $\Rightarrow$  FPGA
- Wireless transmitter-receiver system
- Direct Sequence Spread Spectrum (DSSS) technology

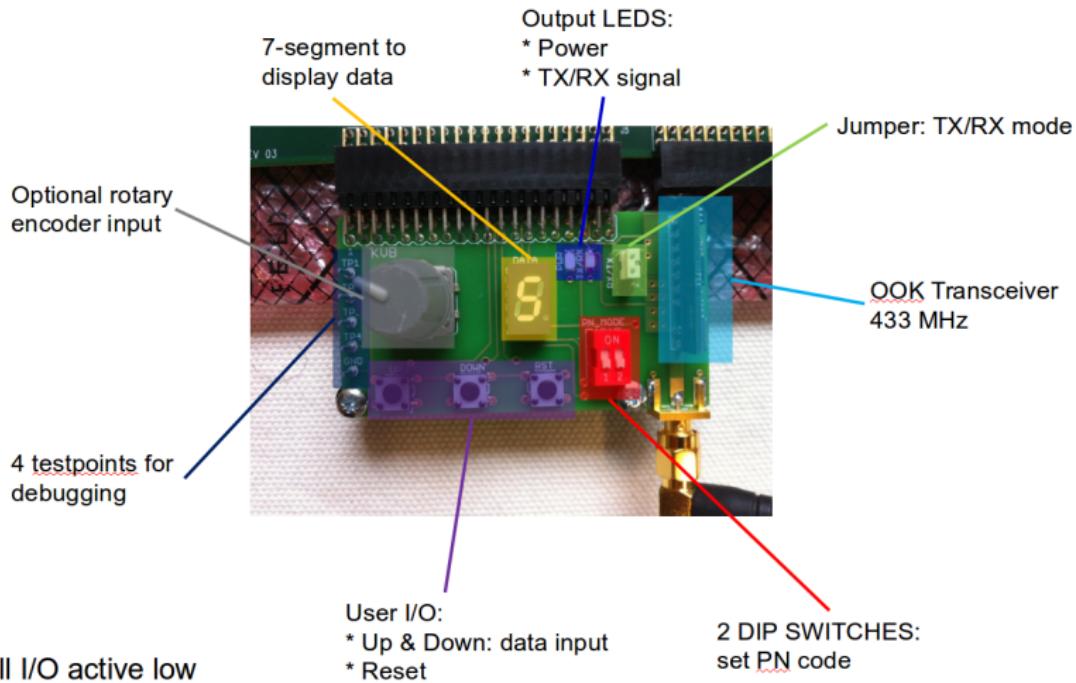
```
LIBRARY IEEE;
USE IEEE.STD.TEXT.TEXTIO.ALL;
USE IEEE.STD.TEXT.ARITH.ARITH_ALL;
USE IEEE.STD.TEXT.LOGIC_UNSIGNED_UNSIGNED_ALL;
USE IEEE.STD.TEXT.LOGIC_UNSIGNED_UNSIGNED_ALL;
ENTITY up_down_counter IS
  PORT (
    clk: IN STD_LOGIC;
    rst: IN STD_LOGIC;
    clk_en: IN STD_LOGIC;
    up: IN STD_LOGIC;
    down: IN STD_LOGIC;
    count_out: OUT STD_LOGIC_VECTOR(3 DOWNTO 0) -- voor seq_seq_dec en data_reg
  );
END up_down_counter;
ARCHITECTURE behav OF up_down_counter IS
  SIGNAL pres_count, next_count: STD_LOGIC_VECTOR(3 DOWNTO 0);
BEGIN
  pres_count <= pres_count;
  SYN_COUT: PROCESS(CLK)
  BEGIN
    IF (RISING_EDGE(CLK) AND CLK_EN = '1') THEN
      IF (UP = '1') THEN
        pres_count := OTHERS &gt; '0';
      ELSE
        pres_count := next_count;
      END IF;
    END IF;
  END PROCESS SYN_COUT;
  COM_COUT: PROCESS(pres_count, UP, DOWN)
  BEGIN
    IF (UP = '1') THEN
      next_count <= pres_count + "0001";
    ELSIF (DOWN = '1') THEN
      next_count <= pres_count - "0001";
    ELSE
      next_count <= pres_count;
    END IF;
  END PROCESS COM_COUT;
END behav;
```



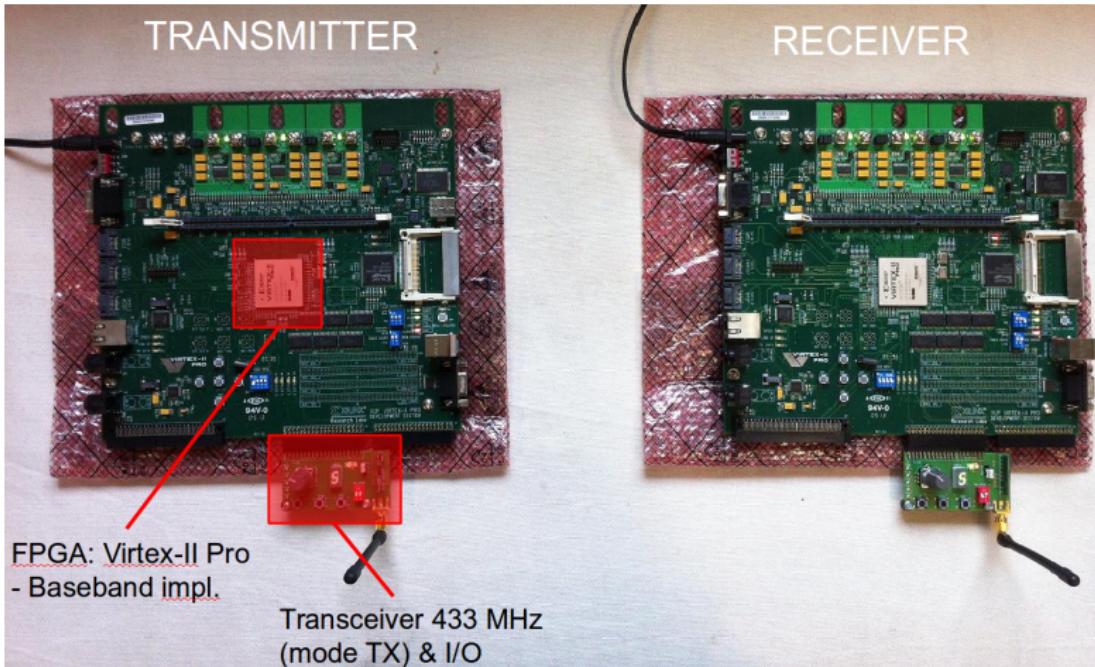
# 1 Spread Spectrum Wireless Communication



# 1 Hardware: transceiver module



# 1 Demo - setup



## 2 Outline

① Introduction

② DSSS

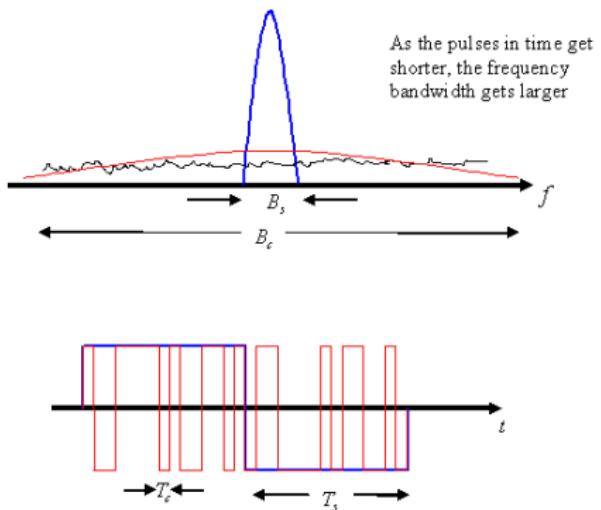
③ Assignment

## 2 DSSS

- ▶ Direct Sequence Spread Spectrum
  - Originally developed for military purposes
  - GPS systems
  - GSM
  - Code Division Multiple Access (eg wifi)

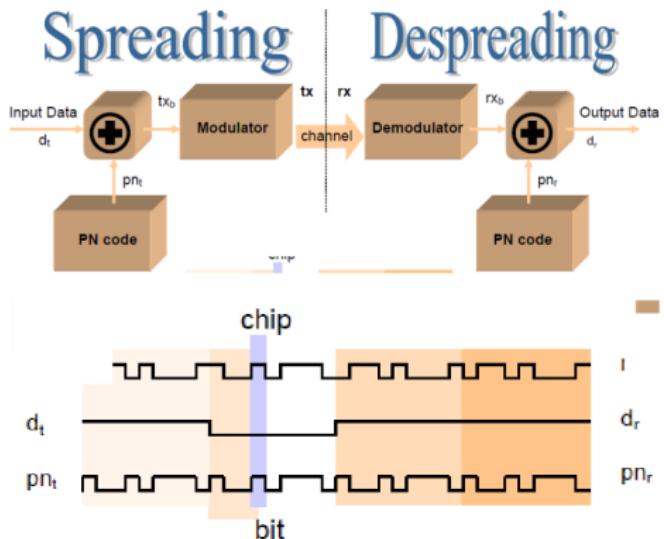
## 2 DSSS - What?

- ▶ Traditional RF: send on 1 frequency
  - disadvantages:
    - Easily traceable
    - Jamming
- ▶ DSSS = spread frequency spectrum
  - advantages:
    - more difficult to trace
    - Jamming is difficult
    - More users per channel (different PN code)
    - Possibly under noise floor



## 2 DSSS - How?

- ▶ XOR input data with PN code
  - XOR = Conditional inverter
  - 2 times XOR'ed: data = original data
- ▶ Data: #bits = packet
- ▶ Every bit XOR with PN code: chips
  - Chips have higher frequency
- ▶ Note: we use 3 possible PN codes



### 3 Outline

① Introduction

② DSSS

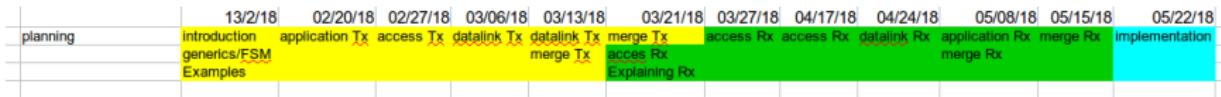
③ Assignment

### 3 Assignment

- ▶ Prepare demo!
  - Send 4 bit binary word using DSSS technology
  - Input on TX with up/down pushbuttons, readout on 7-seg
  - Output on RX using 7-seg

### 3 Workflow

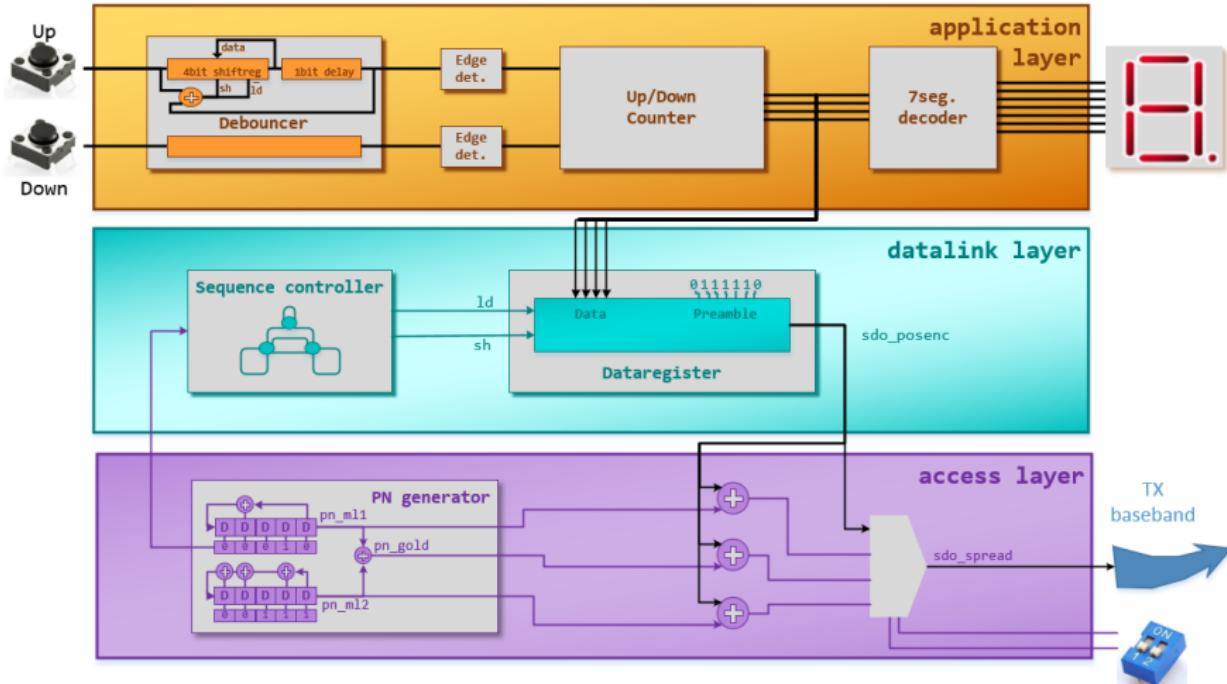
- ▶ Modular/layered design
  - Write VHDL code                      } Modelsim
  - Simulation of VHDL code          }
  - synthesize VHDL code                } Xilinx ISE
  - Bit-file download                    } Xilinx Impact
- ▶ One project (with folder structure)
- ▶ Guiding timeline (approximate)



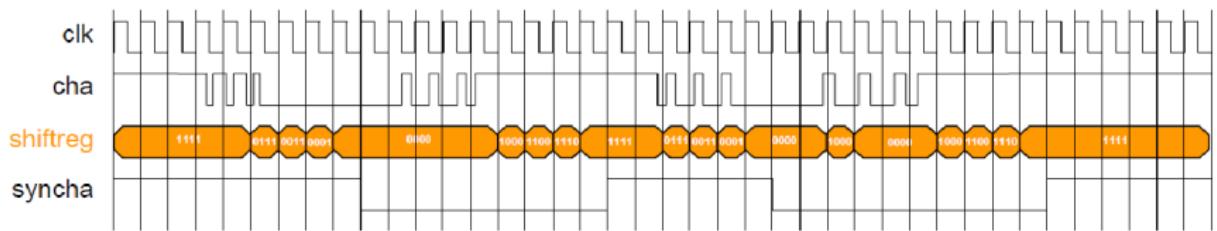
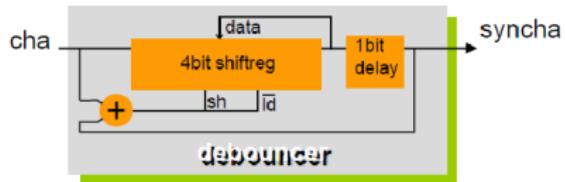
### 3 Evaluation

- ▶ Permanent evaluation:
  - Progress: Show simulation of every layer
  - Please contact me with questions and problems
    - either in class or via [lukas.rondelez@kuleuven.be](mailto:lukas.rondelez@kuleuven.be)
  - Make use of two process method
    - See PDF (fsm.pdf):  
[http://telescript.denayer.wenk.be/kvb/Labo\\_Digitale\\_Synthese/](http://telescript.denayer.wenk.be/kvb/Labo_Digitale_Synthese/)
  - Work modular: testbench for each block
    - More on testbenches in a future session
  - Add comments (**will be quoted**)!
  - Take backups (own responsibility)
  - Simulations run from USB stick are slower
  - Walkthrough with counter (telescript)

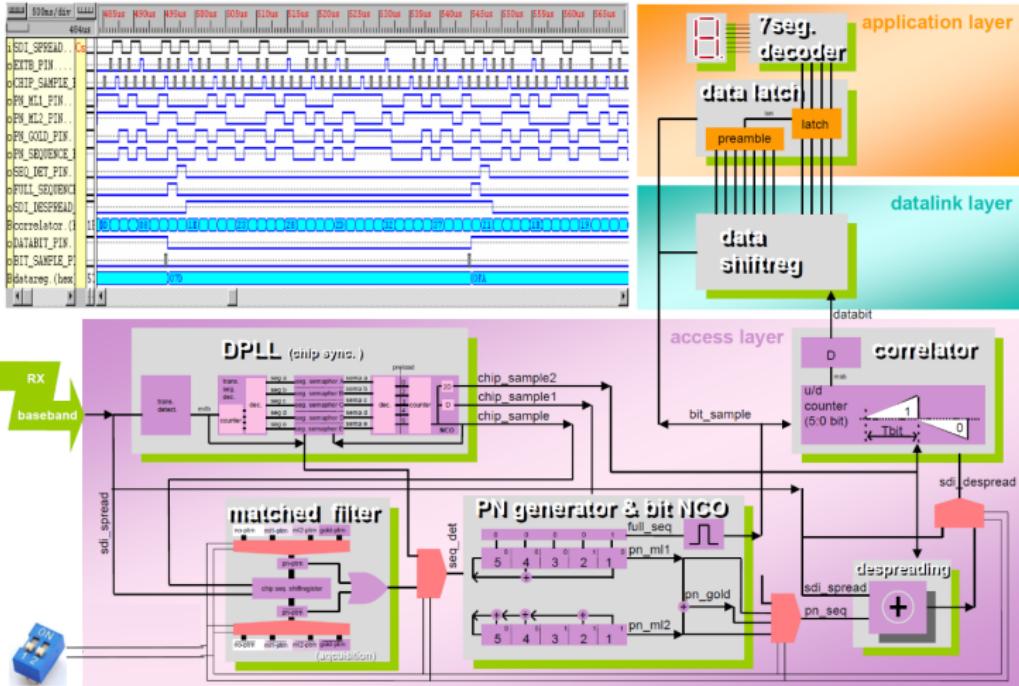
### 3 Spread Spectrum TX: Baseband



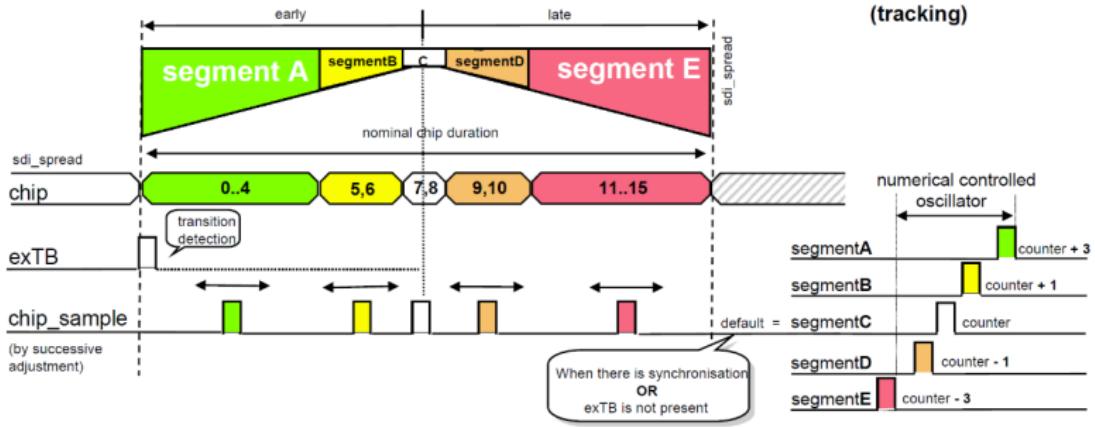
### 3 Spread Spectrum TX: Debouncer



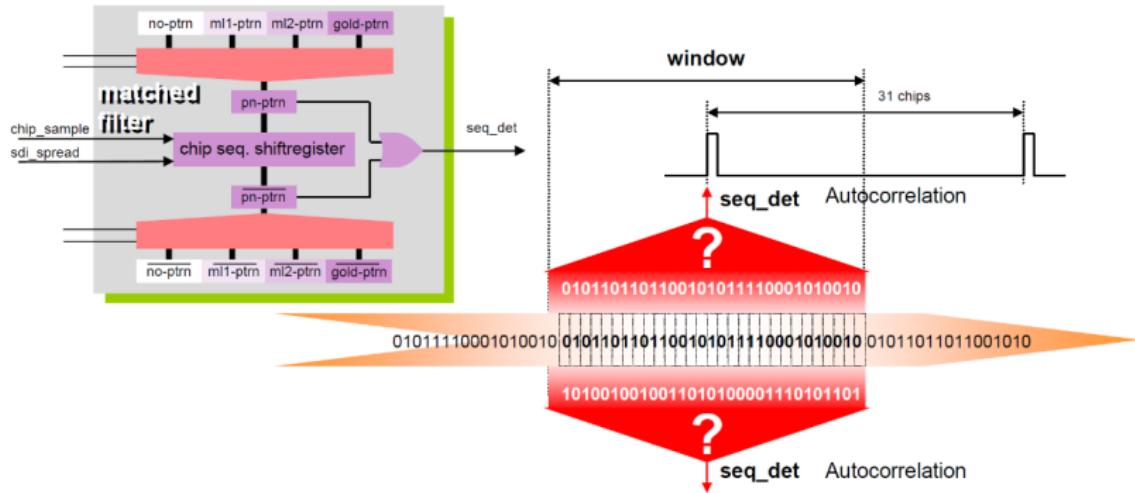
### 3 Spread Spectrum RX: Baseband



### 3 Spread Spectrum RX: Chip Synchronisation



### 3 Spread Spectrum RX: PN Matched Filter



### 3 Spread Spectrum RX: Correlator

