Karlsruher Institut für Technologie

Implementation of a DRM+ transmitter in the GNU Radio software radio framework

Felix Wunsch
Mentor: Jens Elsner

Communications Engineering Lab
Prof. Dr.rer.nat. Friedrich K. Jondral

Outline

- Introduction
- DRM Standard
- Implementation in GNU Radio
- Performance
- Demonstration
- Conclusion

Introduction

- Why digital radio?
  - News and data streams possible
  - Multiple services on one channel
  - Better performance in low SNR environments
- DRM: Digital Radio Mondiale
  - Standard for transmission below 30 MHz (DRM30)
  - Inaugural transmission took place in 2003
  - DRM+ was added to support 30 MHz - 174 MHz band in 2009
- First open source implementation of DRM+

Introduction – Development Process

- Understand the standard
  - Design transmitter and its inverse in MATLAB
- Test against Dream
  - Implements DRM30 receiver
  - Developed at the TU Darmstadt
- Test against commercial DRM receiver
- Port to GNU Radio
- Create unit tests and optimize performance

DRM Standard

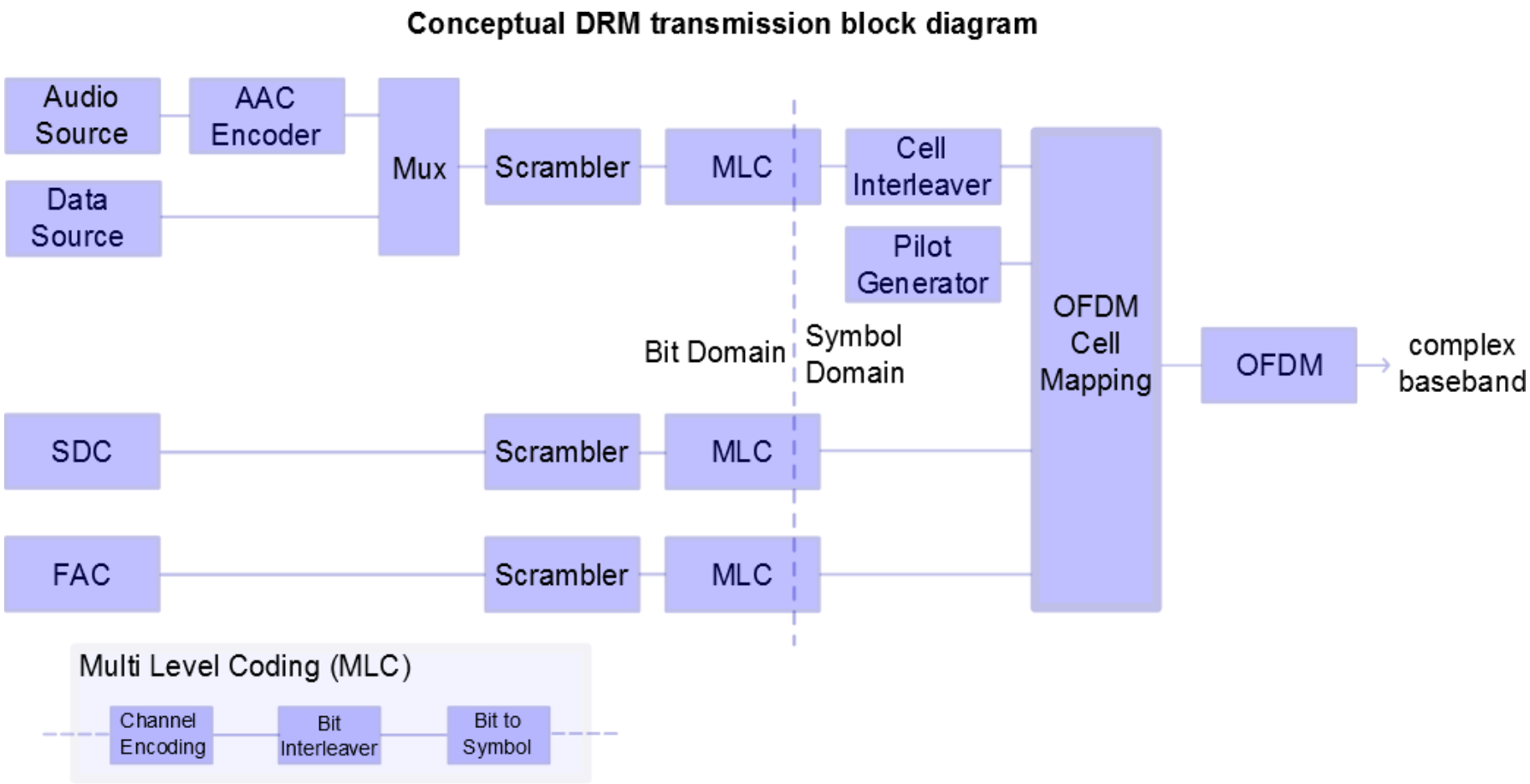
- OFDM system
- Variable data rate, bandwidth and error robustness
- Subset of MPEG-4 standard used for audio coding
  - AAC, CELP, HVXC
- Three logical channels
  - Main Service Channel (MSC)
  - Service Description Channel (SDC)
  - Fast Access Channel (FAC)
- All stages‘ parameters tightly intertwined to optimize performance

DRM Standard – Physical Layer

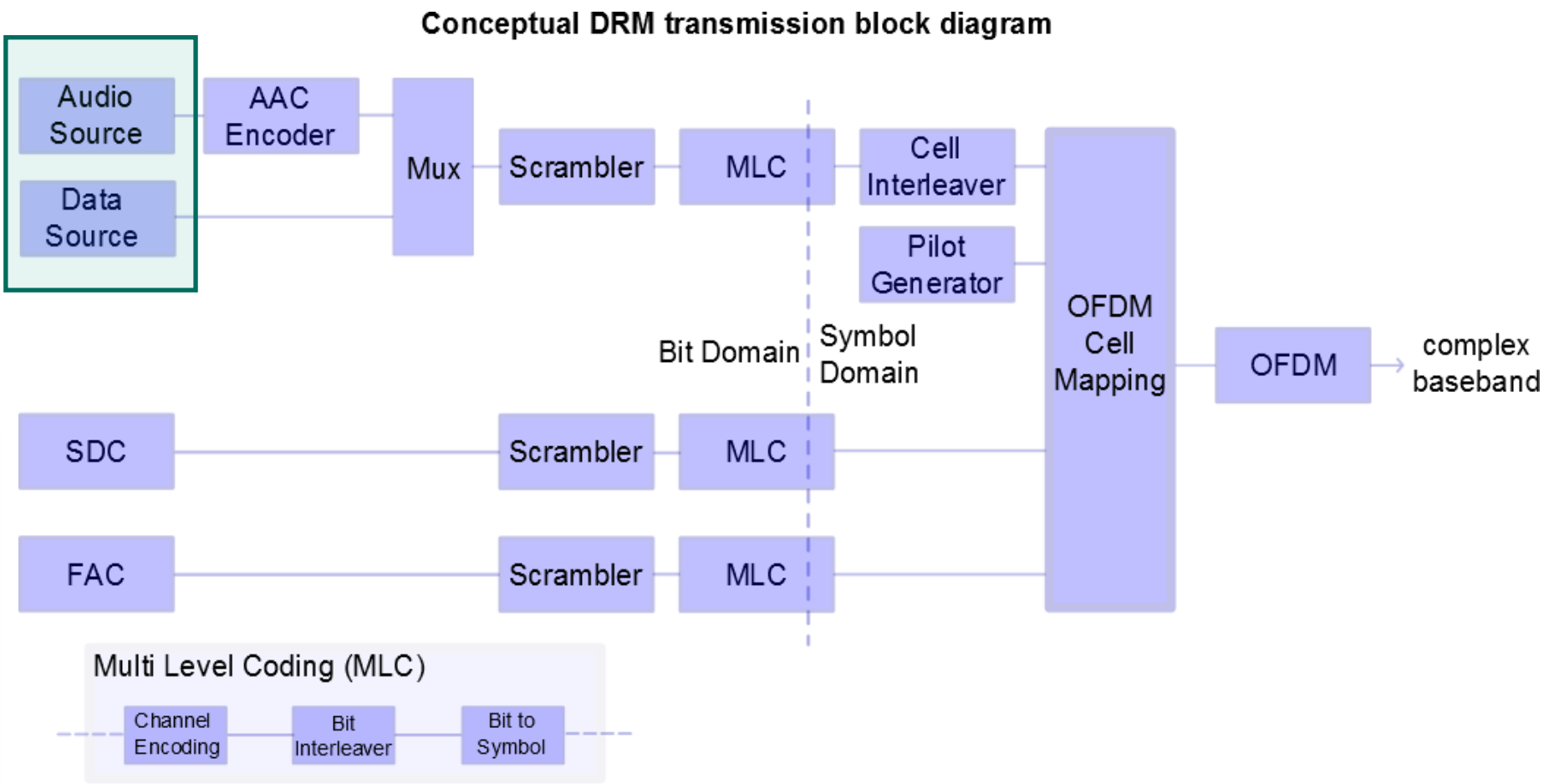
- Two main parameters:
  - RM: Robustness Mode (A - E, error robustness)
  - SO: Spectrum Occupancy (0 - 5, occupied bandwidth)

	DRM30	DRM+
Bandwidth	4.5 kHz … 20 kHz	100 kHz
Symbol Duration	9.33 ms … 24 ms	2.25 ms
Guard Interval	2.66 ms … 7.33 ms	0.25 ms
Bit Rate	4.8 kbps … 72 kbps	37.3 kbps … 186.4 kbps
Frame Duration	400 ms	100 ms

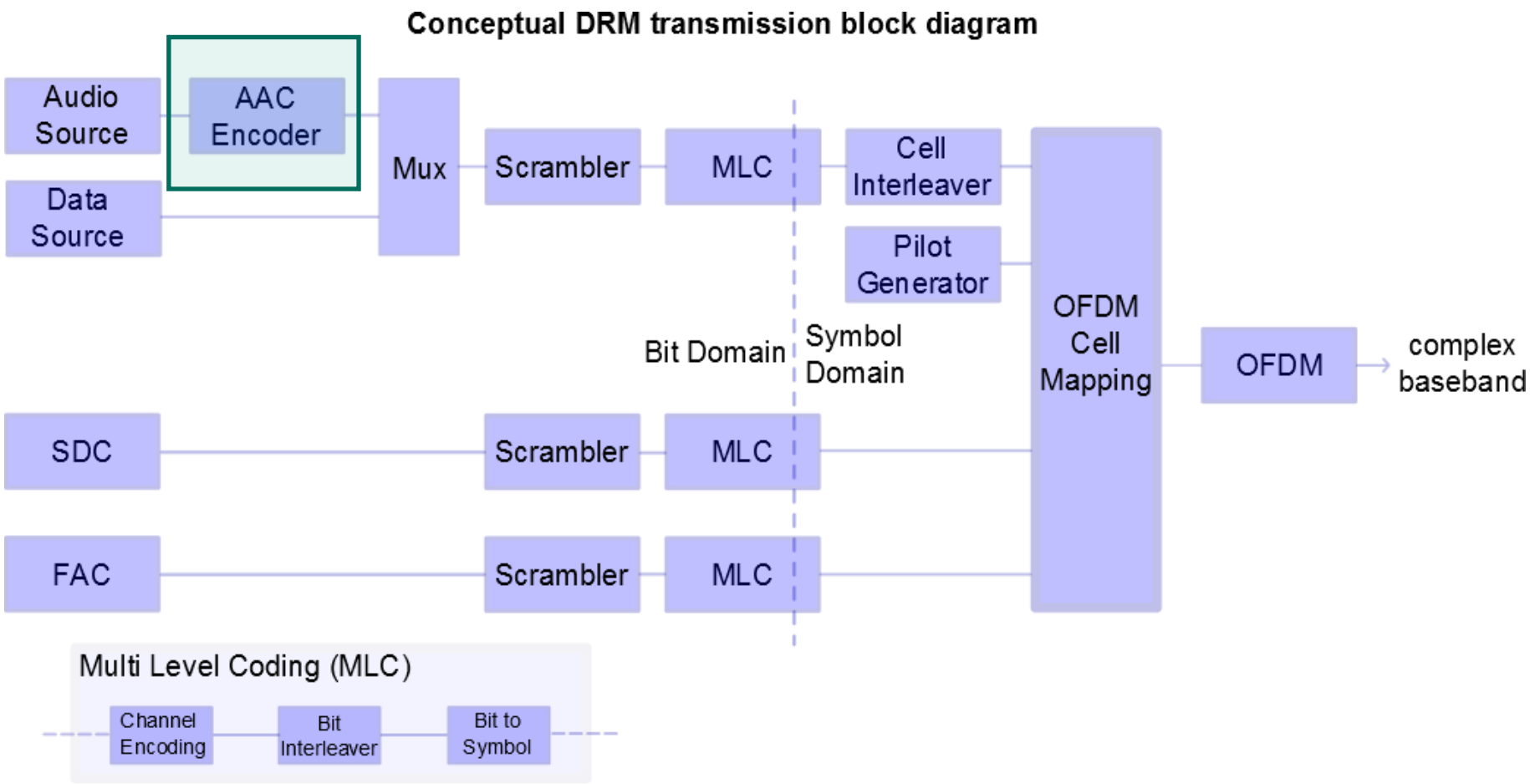
DRM Standard – Flow graph



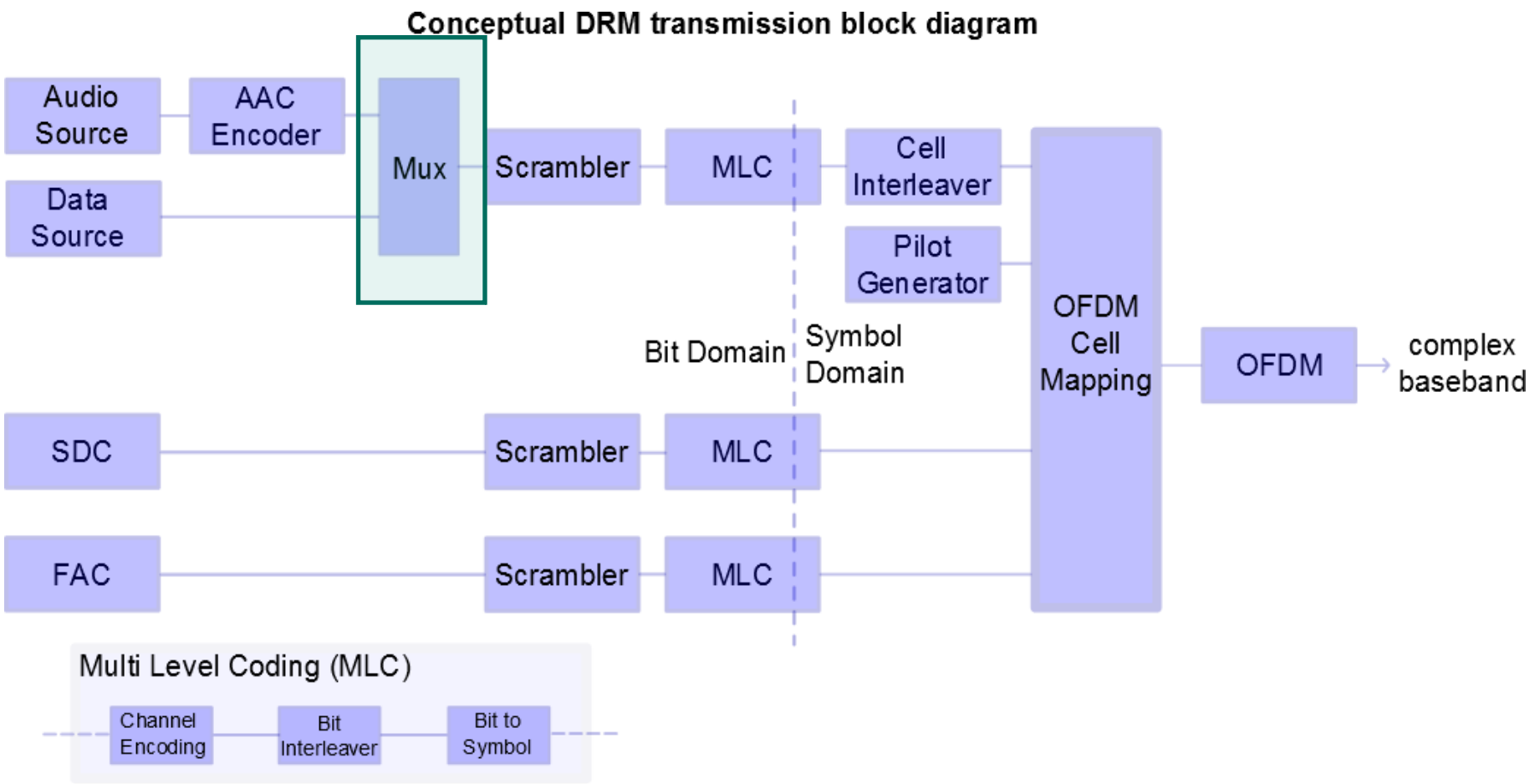
DRM Standard – Flow graph (cont.)



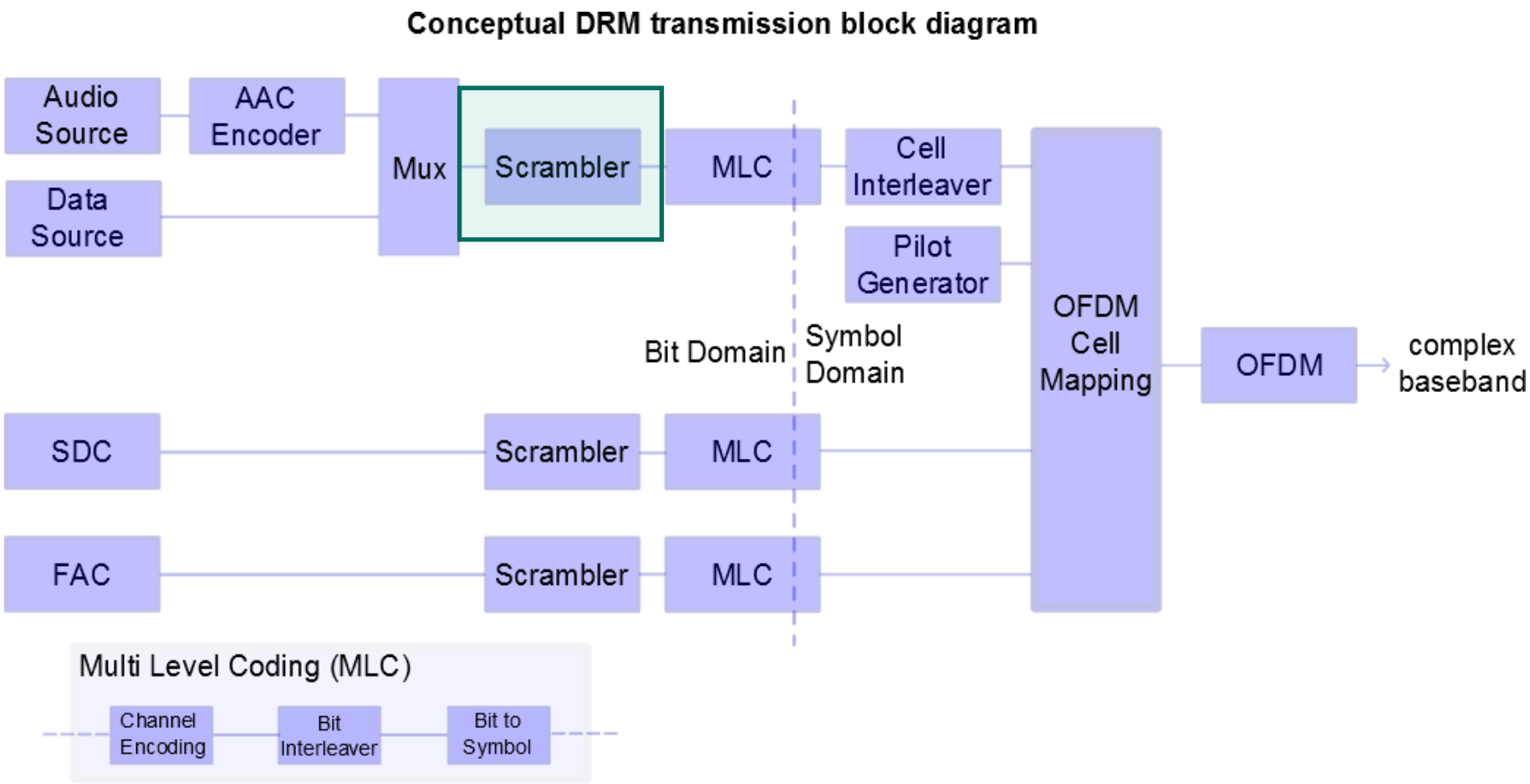
DRM Standard – Flow graph (cont.)



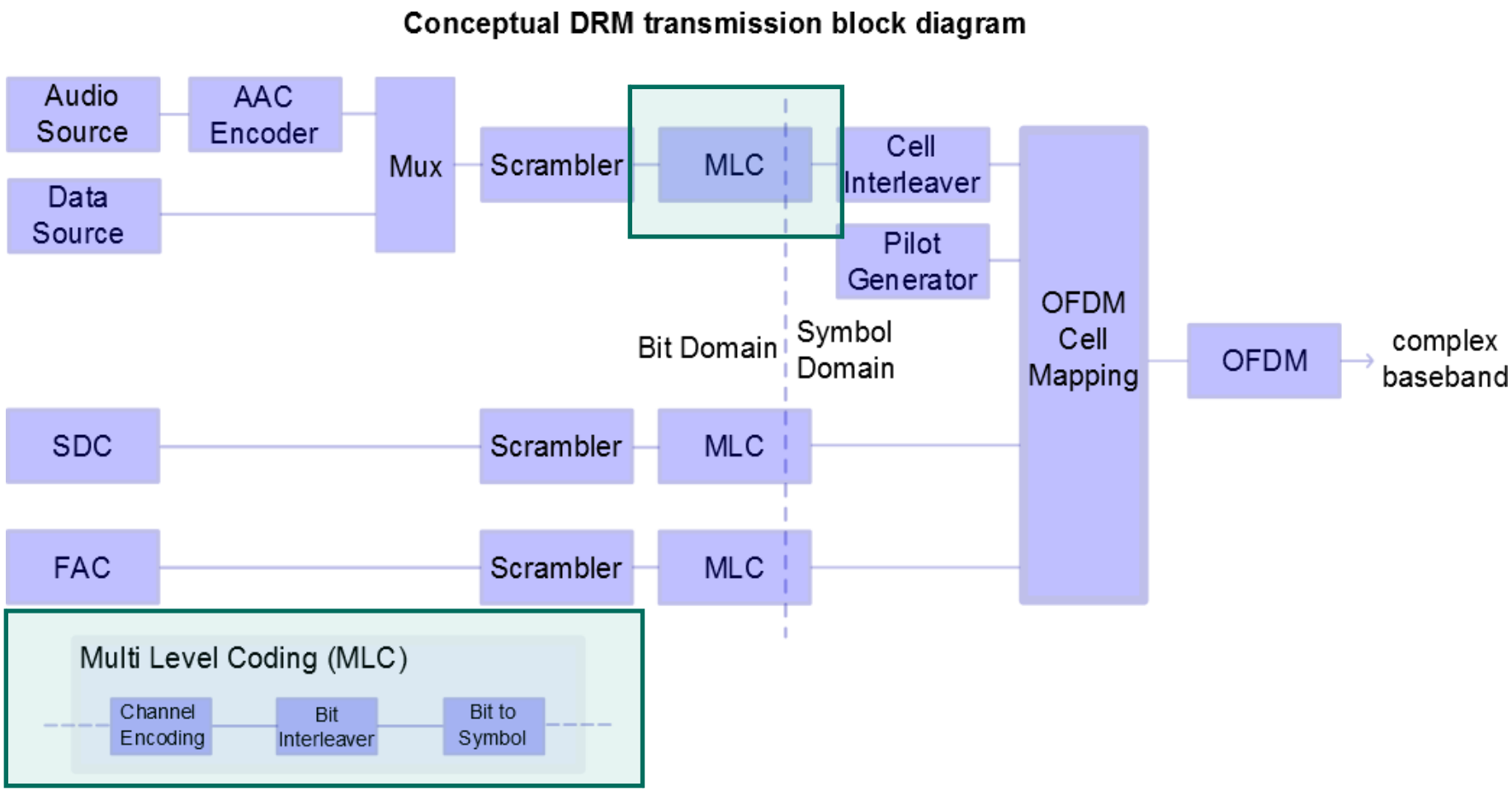
DRM Standard – Flow graph (cont.)



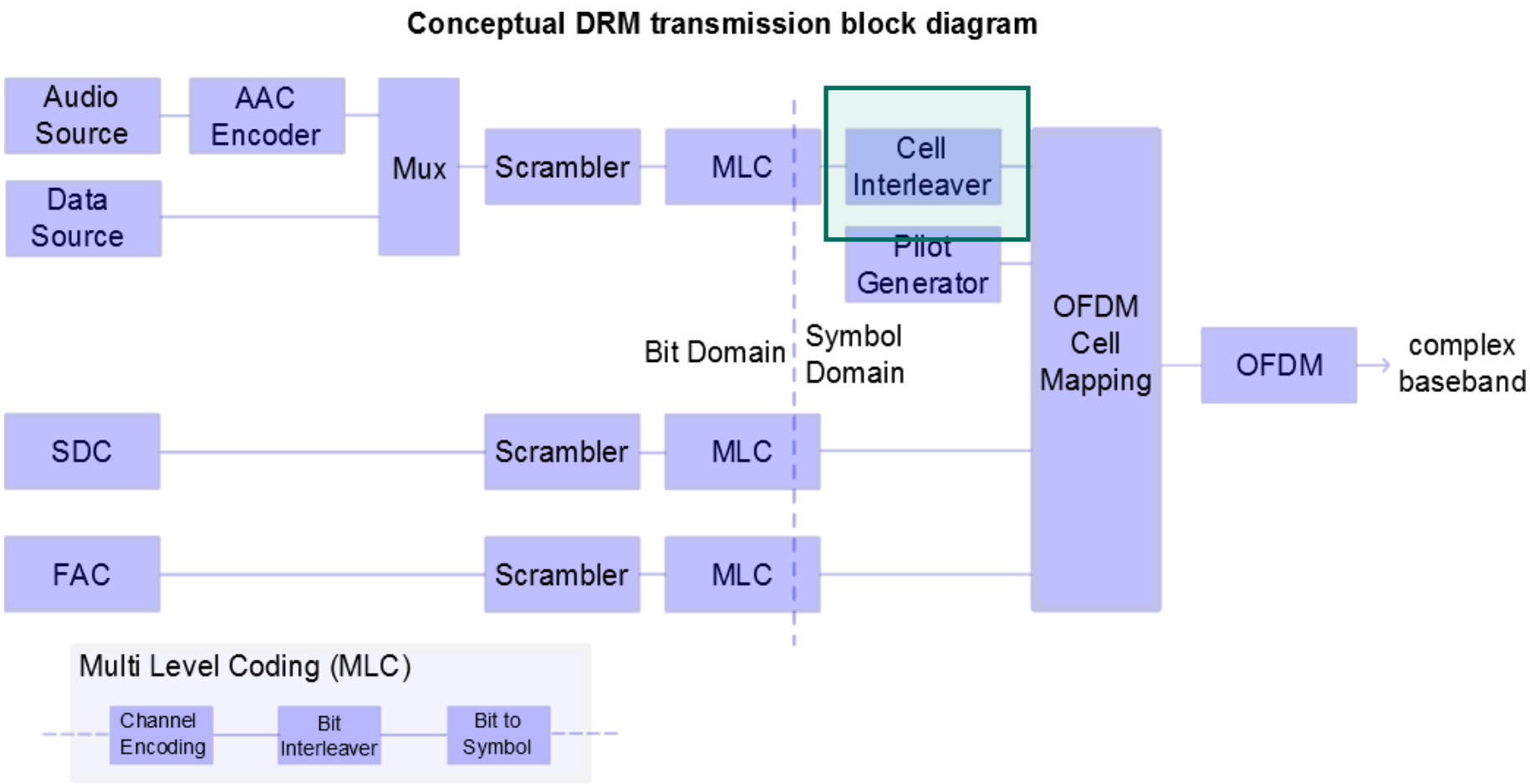
DRM Standard – Flow graph (cont.)



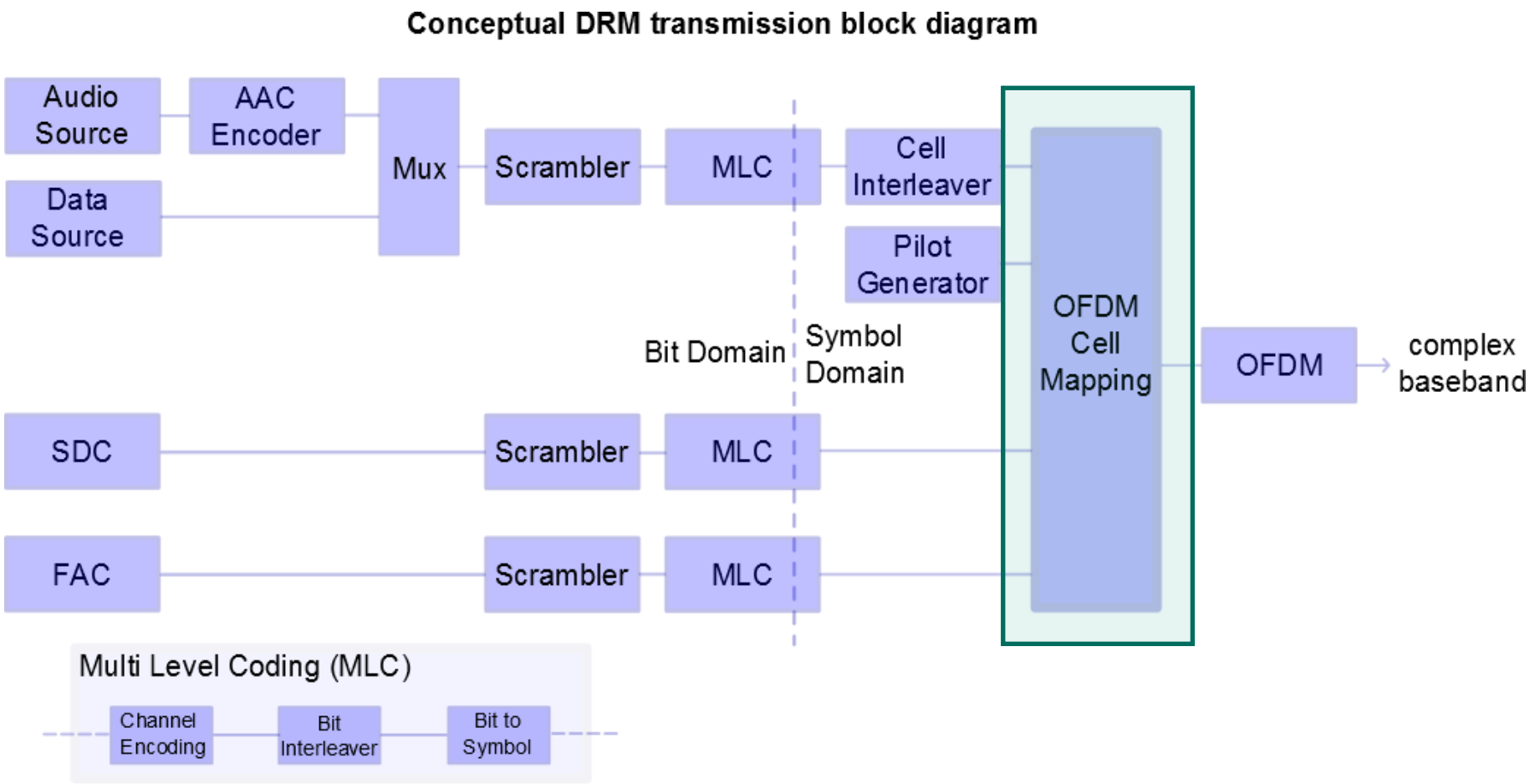
DRM Standard – Flow graph (cont.)



DRM Standard – Flow graph (cont.)

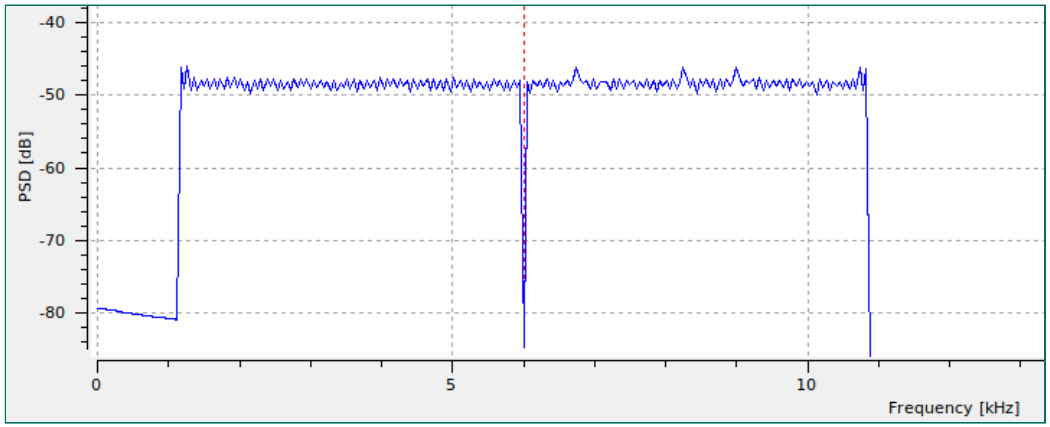


DRM Standard – Flow graph (cont.)

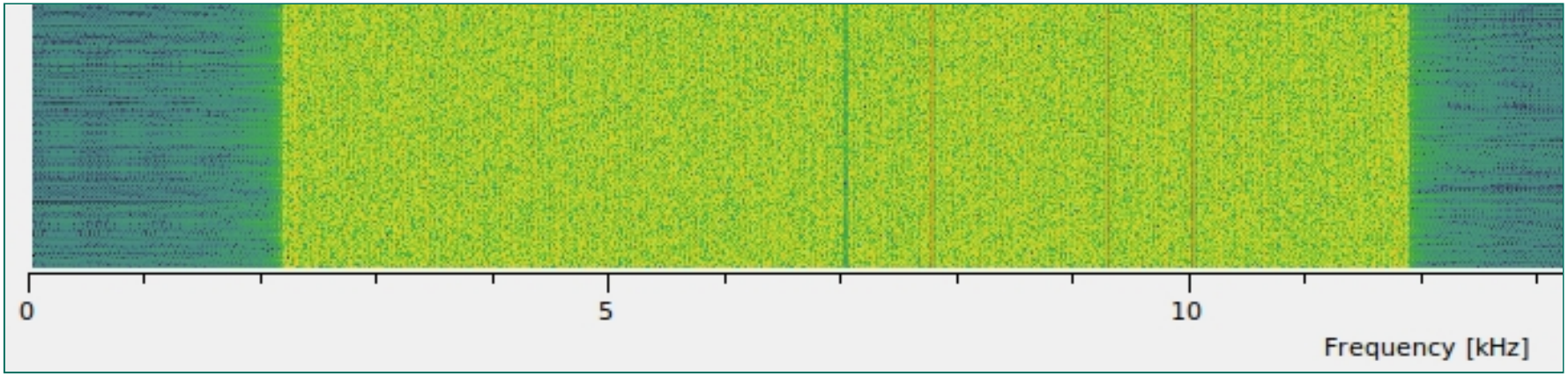


DRM Standard – Flow graph (cont.)

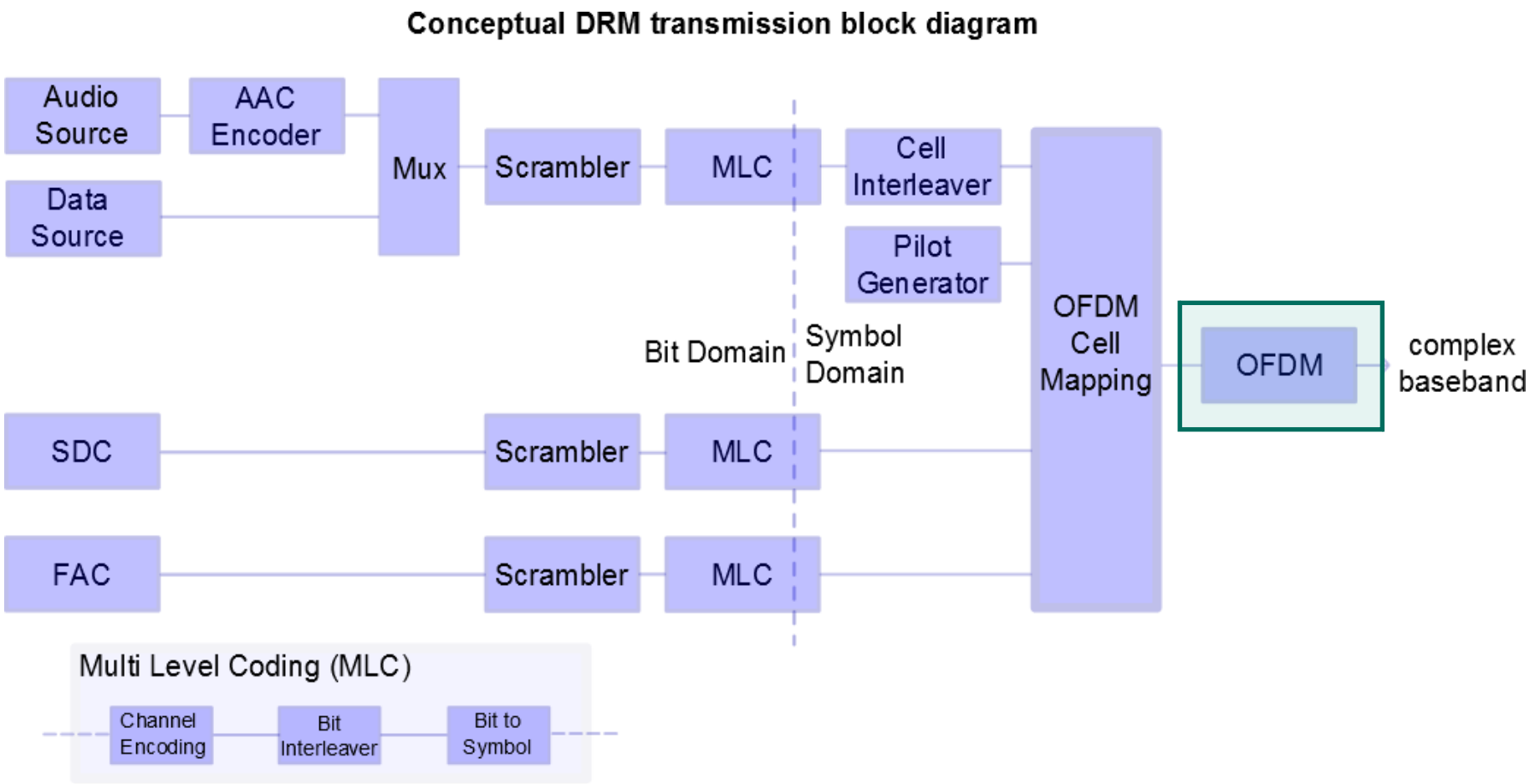
- Different pilot cells
  - Frequency reference cells
  - Time reference cells
  - Gain reference cells
  - AFS reference cells



- Variable no. of subcarriers
- Variable carrier spacing

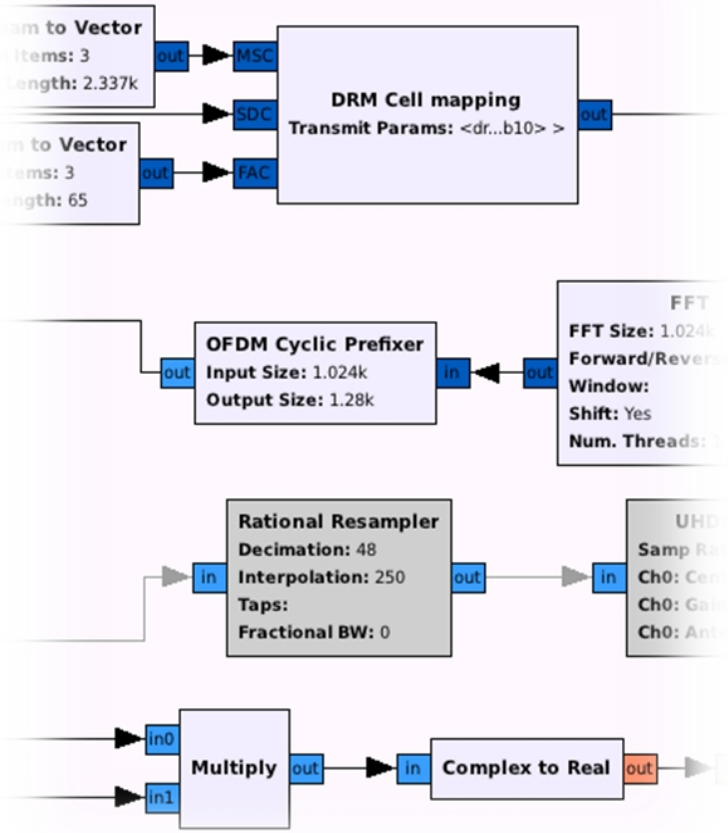


DRM Standard – Flow graph (cont.)



Implementation in GNU Radio

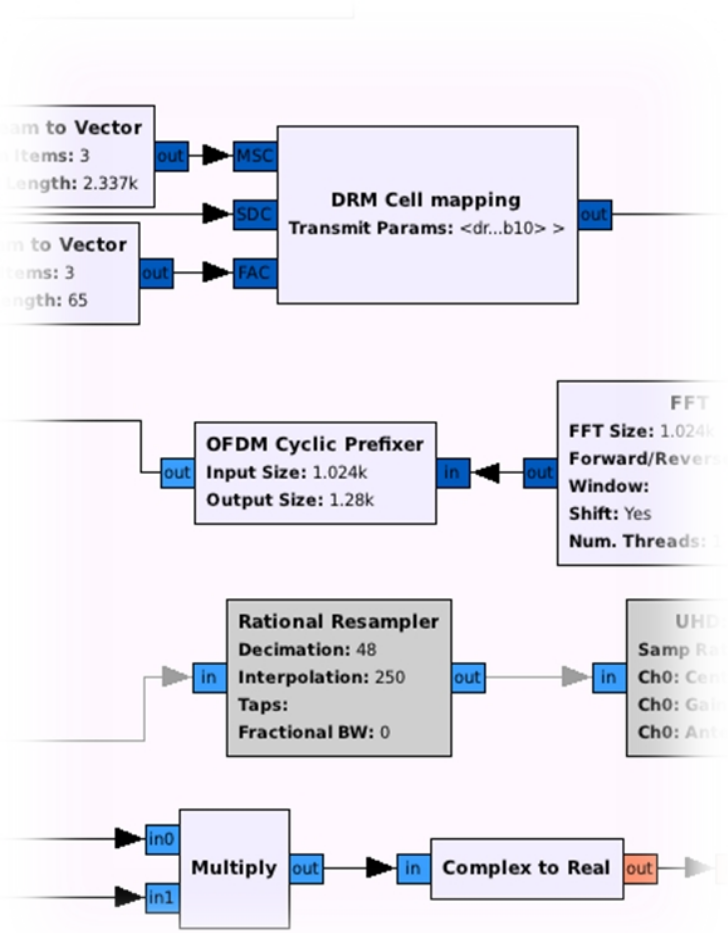
- New module „gr-drm“
- Fully integrated into GRC
  - No deep knowledge of DRM required
  - Flow graphs can be used as-is
- As modular as possible
  - Generic blocks
    - Puncturing
    - Interleaver
    - …
- http://github.com/kit-cel/gr-drm



Implementation in GNU Radio – Wish List

- Reload XML definitions
  - Saves time

- Hier block transparency
  - More depth



Performance

Test environment
Intel i5-2520M
GNU Radio 3.6.1
Ubuntu 12.04 (64 bit)
USRP N210 with BasicTX

Transmitter configuration
RM A, SO 3 (10 kHz), 64 QAM
26.56 kbps (AAC mono)

- Real-time capable
- Low CPU consumption
- Suboptimal buffer allocation
- AAC encoder library dominates CPU consumption

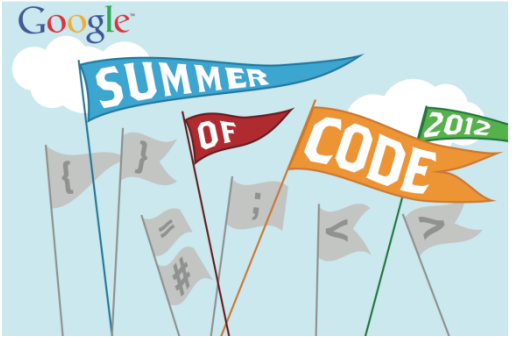
```
top - 16:25:11 up 5 min,  4 users,  load average: 0.64, 0.70, 0.34
Tasks: 160 total,   3 running, 157 sleeping,   0 stopped,   0 zombie
Cpu(s):  4.1%us,  5.1%sy,  0.0%ni, 90.6%id,  0.0%wa,  0.0%hi,  0.2%si,  0.0%st
Mem:   8058848k total,  1676924k used,  6381924k free,    78624k buffers
Swap:  8267772k total,        0k used,  8267772k free,   952892k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 2323 felixwun -30   0 3172m 298m 255m S   19  3.8   3:06.79 python
```

Demonstration

- Signal generation with gr-drm GRC flow graph
- Decoding with NEWSTAR DR-111 (live)
- Parameters can be changed directly in GRC
- Configuration: 10 kHz bandwidth, RM A, 24 kHz wav-file, 64-QAM

Conclusion

- Great GSoC experience
  - Over 5000 lines of code
  - Daily interaction with mentor
  - Very helpful community



- DRM30 transmitter has been implemented in GNU Radio
  - DRM+ is implemented but untested

- Next steps:
  - Integration of Fraunhofer AAC encoder
  - Implementation of DRM Receiver in GNU Radio

- DRM+ might become the new standard for digital broadcasting in Brazil

ETSI ES 201 980 V3.1.1 (2009-08)

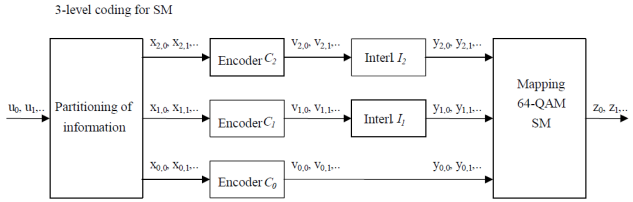
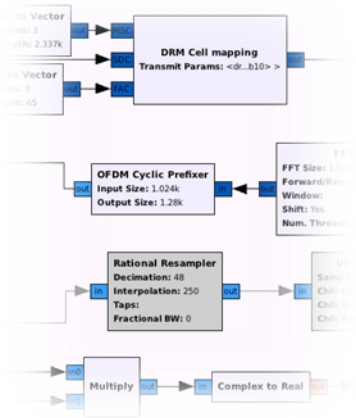


Figure 29: Multilevel coding with 3 levels for SM



Q & A