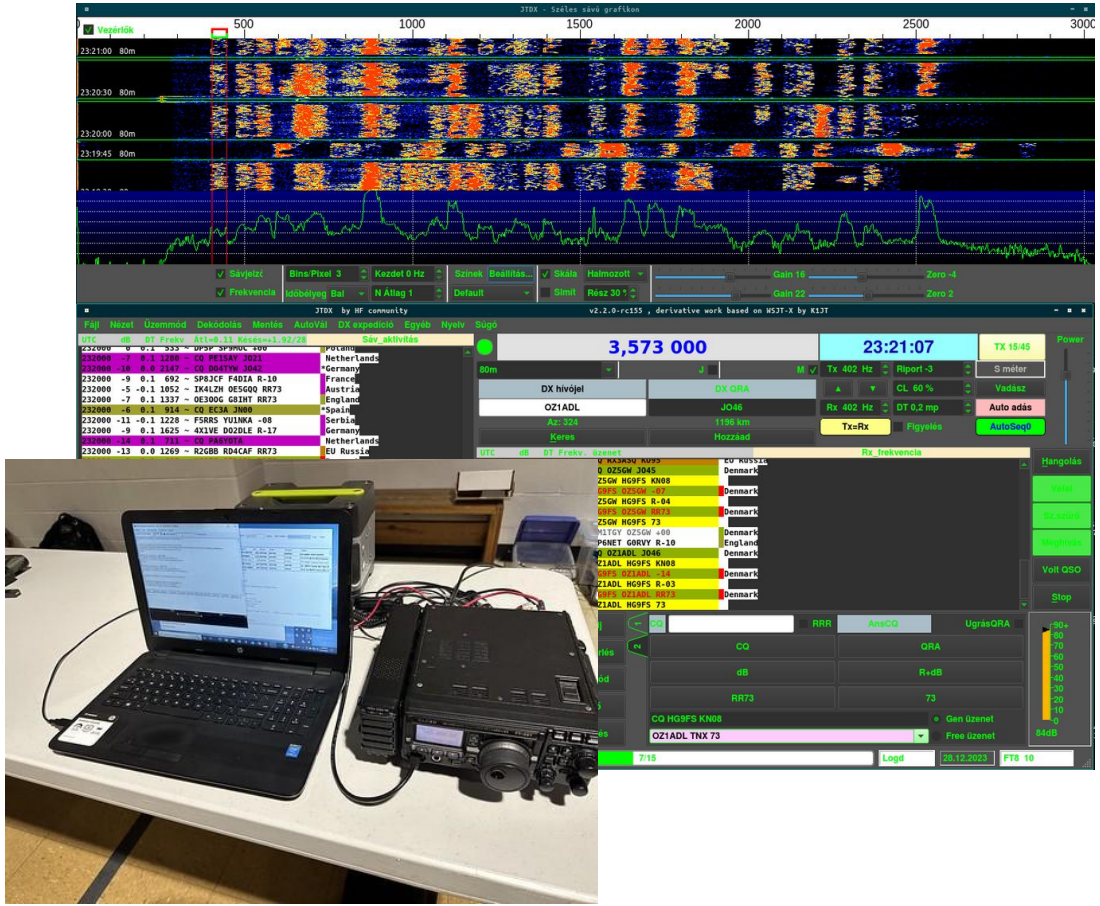
FT8 Encoder

Hisen Zhang
Adv VLSI, Spring 2024

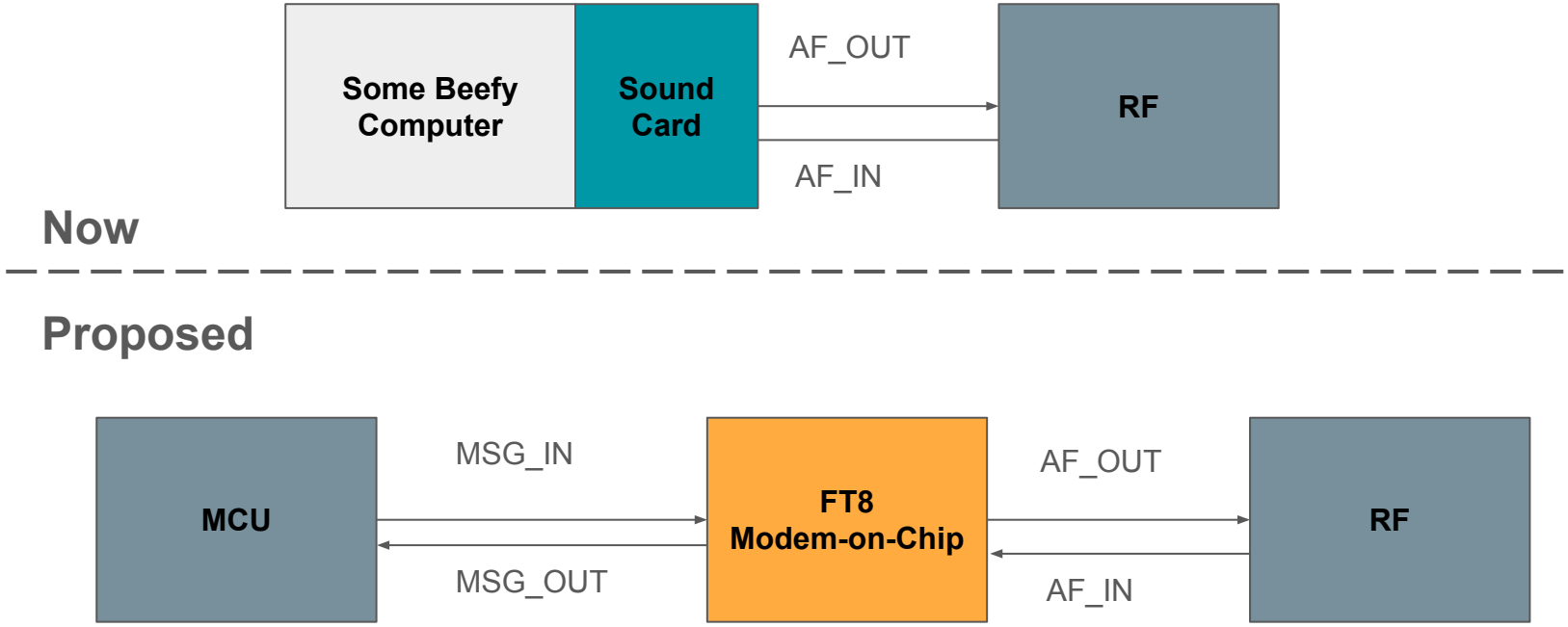
What is FT8

Ham Radio Digital Mode

- 15 seconds time slots
  - 12.6 sec Tx
- 50 Hz bandwidth
  - 5.86 Hz tone spacing
  - 6.25 baud
  - AFSK-8 modulation
- 77-bit Payload
- Forward error correction
- Domain encoding
  - message format
  - call signs



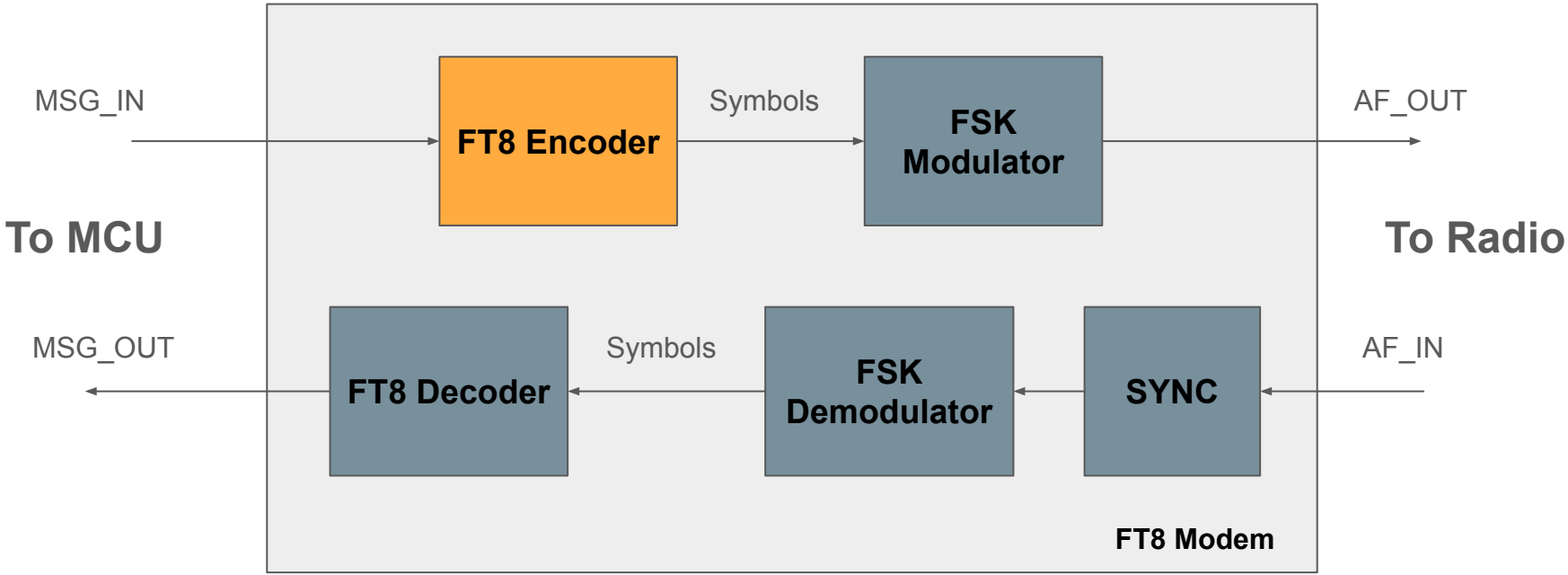
FT8 Modem-on-Chip



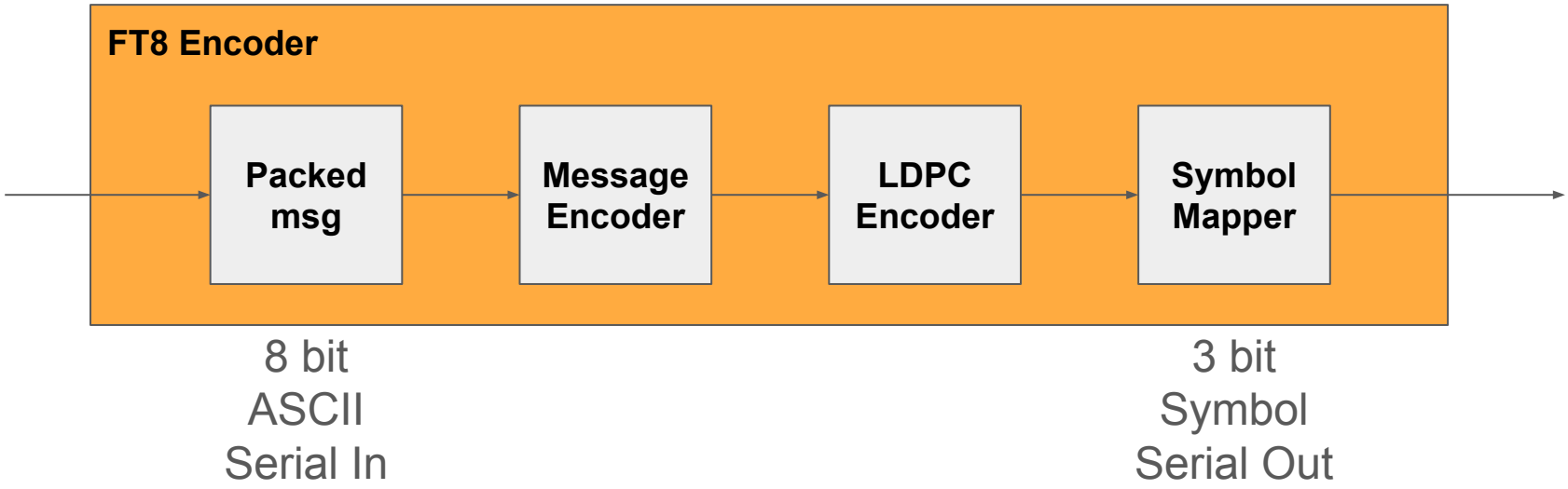
Motivation

- Current implementations are all software
  - Need a powerful PC
  - Not possible under some case
- Protocol is not mature and not active evolution
- Can interface with microcontroller
- More power efficient

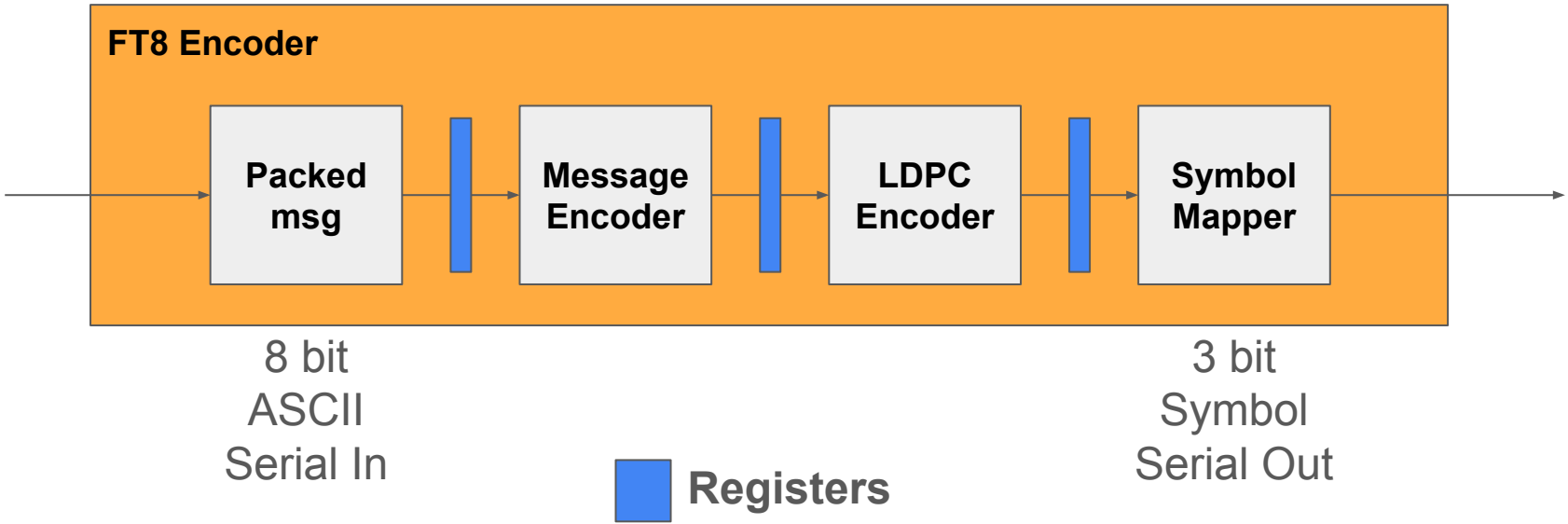
FT8 Modem-on-Chip



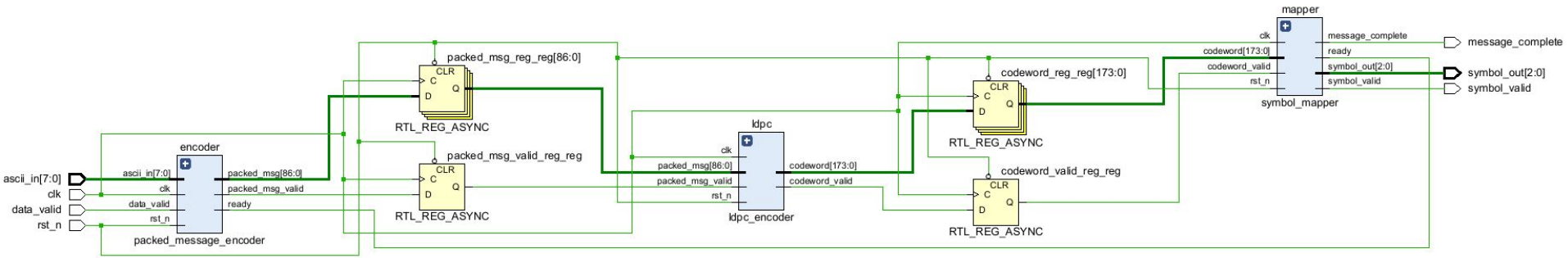
System Diagram



Pipelining


FT8 Encoder
Packed msg
Message Encoder
LDPC Encoder
Symbol Mapper
8 bit ASCII Serial In
3 bit Symbol Serial Out
Registers

RTL Design


ascii_in[7:0]
clk
data_valid
rst_n
encoder
ascii_in[7:0]
clk
data_valid
rst_n
packed_msg[86:0]
packed_msg_valid
ready
packed_message_encoder
packed_msg_reg_reg[86:0]
CLR
C
D
Q
RTL_REG_ASYNC
packed_msg_valid_reg_reg
RTL_REG_ASYNC
ldpc
clk
packed_msg[86:0]
packed_msg_valid
rst_n
codeword[173:0]
codeword_valid
ldpc_encoder
codeword_reg_reg[173:0]
RTL_REG_ASYNC
codeword_valid_reg_reg
RTL_REG_ASYNC
mapper
clk
codeword[173:0]
codeword_valid
rst_n
message_complete
ready
symbol_out[2:0]
symbol_valid
symbol_mapper
message_complete
symbol_out[2:0]
symbol_valid

Message Encoder

- input: ASCII characters
- concatenates into a 72-bit message
- appends three zero bits for padding
- calculates a 12-bit CRC
- output: 87-bit packed message

LDPC Encoder

- Takes the 87-bit packed message
- encodes it into a 174-bit codeword
  - using the LDPC (174,87) code.
- using the generator matrix, Nm
  - Defined in the Fortran Implementation
  - See technical report for details

Symbol Mapper

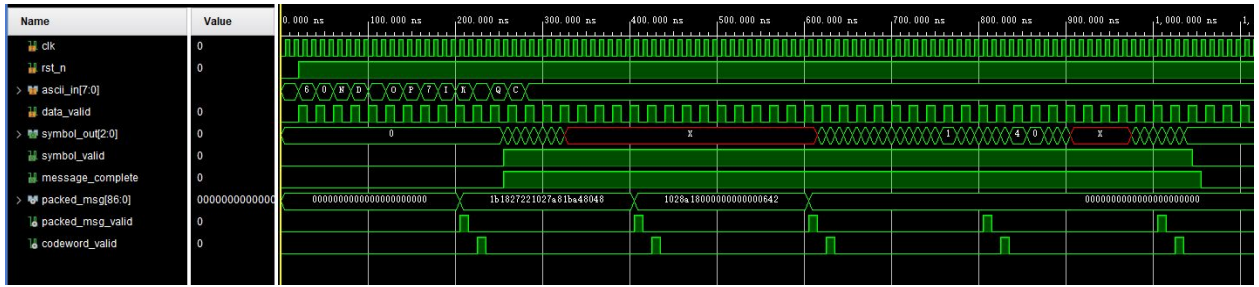
- Divides the input codeword into 58 groups of 3 bits each
- maps each group to an 8FSK symbol.
- inserts special synchronization (Costas arrays)
  - at the beginning, middle, and end of the transmission
- The output: 79 3-bit symbols

CRC

- Packed Message Encoder calculates the 12-bit CRC
- Checksum is then appended to the message
  - resulting in an 87-bit packed message
- CRC Polynomial
  - x^12 + x^11 + x^3 + x^2 + x + 1
  - can be represented in binary as 1100000001111
  - or in hexadecimal as 0xC0F

Evaluation

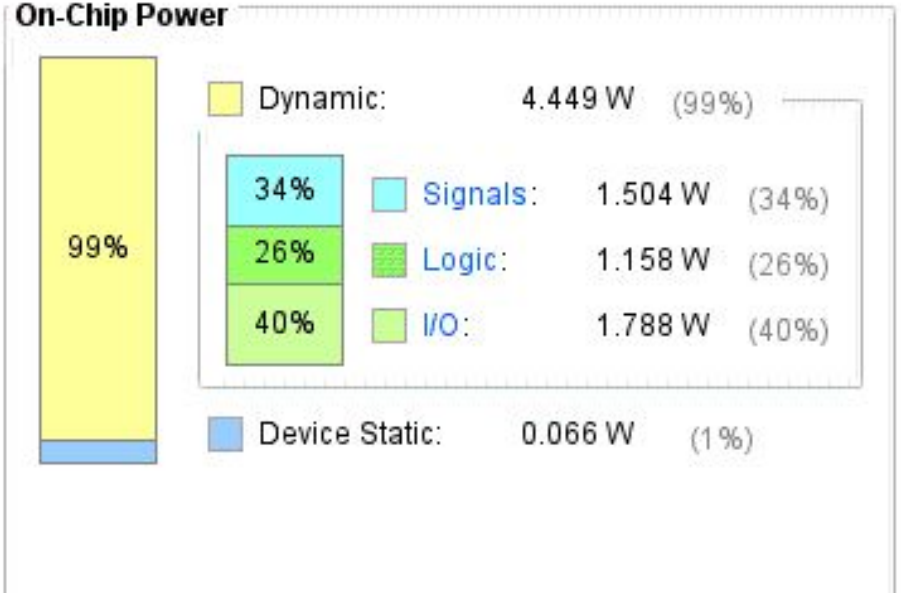
- Input Sample: CQ KI7PO DN06
- It assigns the ASCII value
- asserts the data_valid signal
- symbol_valid signal indicating the validity of output
- Capture!



Power

Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power:	**4.515 W**
Design Power Budget:	**Not Specified**
Power Budget Margin:	**N/A**
Junction Temperature:	**48.8°C**
Thermal Margin:	51.2°C (9.7 W)
Effective ϑJA:	5.3°C/W



Limitation & Future Work

- The current work is only the encoder
  - Need decoder for two-way communication
- Each stage can be further optimized
  - E.g. LDPC encoding can be parallelized
  - Better I/O interface
  - Clock gating
- The pipelining -> smaller stages

Conclusion

- Hardware FT8 encoder implemented with message packing, CRC, LDPC, and symbol mapping
- Pipelined architecture improves throughput and enables parallel processing
- Power-efficient design suitable for low-power applications
- Foundation for future expansion, optimization, and integration towards a complete FT8 modem-on-chip