FPL2012 *22nd International Conference on Field Programmable Logic and Applications*

DATA CODING FUNCTIONS FOR SOFTWARE DEFINED RADIOS IMPLEMENTED ON R3TOS

Raúl Torrego
IK4-IKERLAN
rtorrego@ikerlan.es

IK4 IKERLAN Research Alliance

THE UNIVERSITY of EDINBURGH

MONDRAGON UNIBERTSITATEA | GOI ESKOLA POLITEKNIKOA ESCUELA POLITÉCNICA SUPERIOR

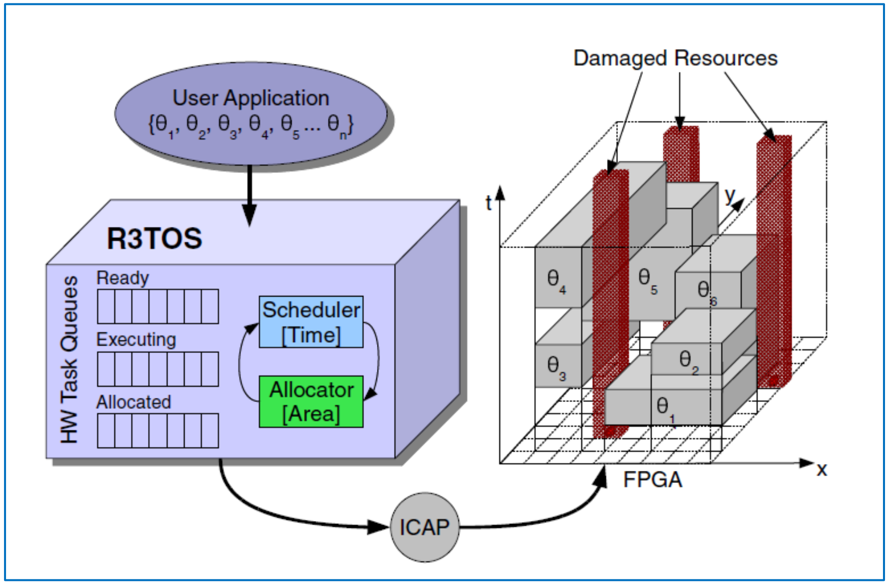
INDEX

- Introduction
  - R3TOS
  - Software Defined Radios
- Implementation
  - Data Coding Functions
  - I2CI: ICAP-based Inter-task Communication Infrastructure
  - Function context saving and restoration procedure
  - Function Parameterization strategy
- Results
- Conclusions and future work
- Questions

R3TOS (overview)

– Reliable Reconfigurable Real-Time Operating System

- Enables HW tasks to behave like SW tasks
- Tasks are are swapped in and out of the FPGA's reconfigurable area (real-time) via partial reconfiguration
- Damaged resources are avoided (scrubbing techniques)



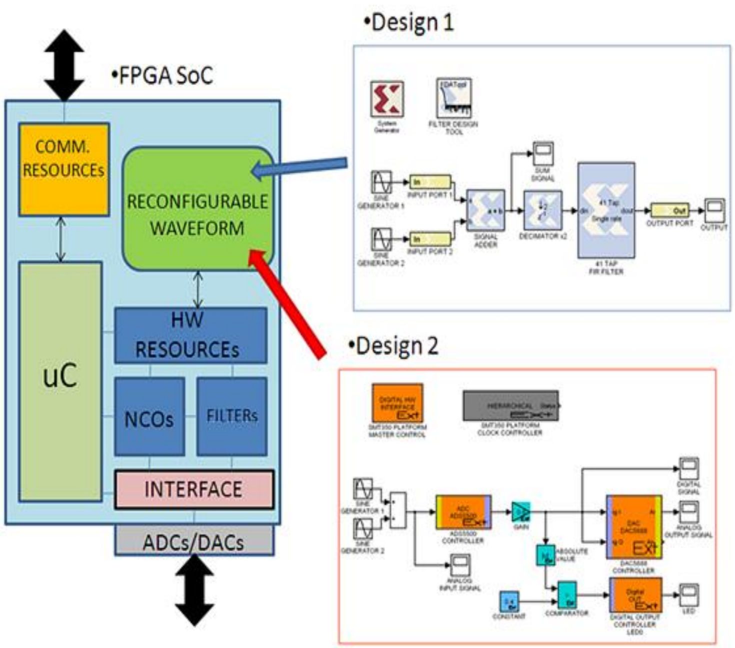
R3TOS (components)

- ICAP controller
  - Efficient control for the Internal Configuration Access Port
  - Composed of a finite state machine and a PicoBlaze microcontroller
  - High level functions: task loading and blanking, data feeding and collection to/from tasks, scrubbing…
  - 32 bit words at 100 MHz
- Scheduler
  - Decides the execution order of the tasks
  - Earliest Deadline First algorithm implemented on a PicoBlaze
  - Preemptive operating system
  - True parallel task execution

INTRODUCTION

- R3TOS (components)
  - Allocator
    - Looks for an appropriate site within the reconfigurable area of the FPGA for the tasks
    - Input: damaged resources, needed resources, occupied resources
    - Empty Area Compaction algorithm implemented on a PicoBlaze
  - Miscellaneous
    - MicroBlaze processor (external communications, bitstream management, task generation)
    - Recovery unit
    - Configuration guardian

Software Defined Radios



- Definition:

  Communication system where a single piece of hardware has <u>different functionalities in different times</u>

- Close relation with FPGA dynamic partial reconfiguration
- High flexibility (change of parameters, change of communication standard...)
- Reliable communications

Fits with R3TOS

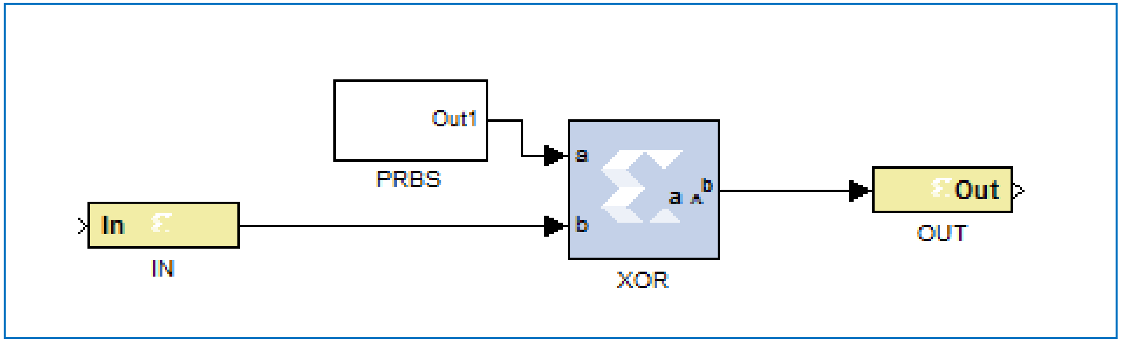
Data Coding Functions

- First tasks in a SDR modulator
  - Present in UMTS, WiFi and WiMAX
  - Proof-of-concept of a future full SDR
- Implemented with System Generator
  - Xilinx's rapid prototyping tool
  - Graphical programming and automatic HDL code generation
- Two different implementation
  - Individual functions (5)
  - Full-standard implementation (single task with all the functions inside)

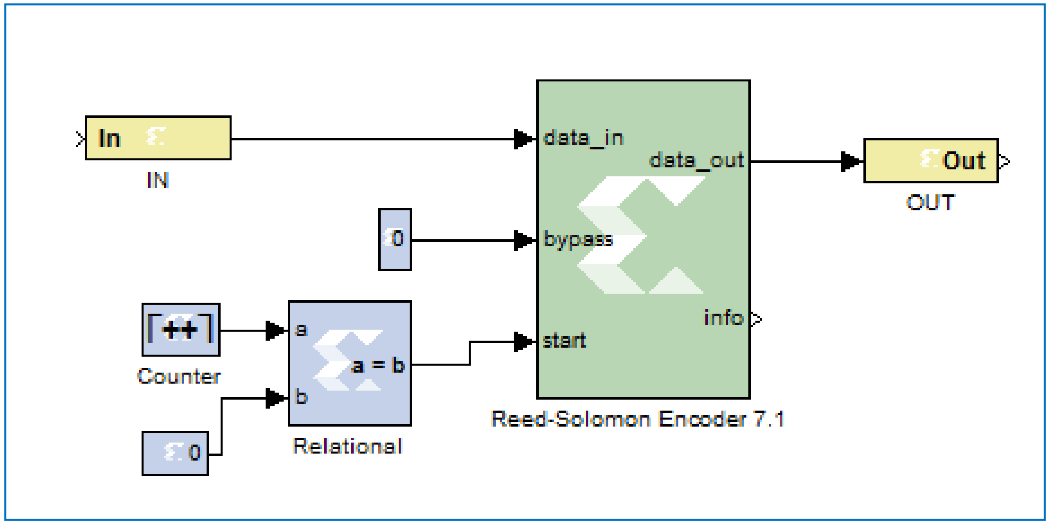
Data Coding Functions

– Data randomizer

- Provides a simple encryption
- Pseudo Random Binary Sequence generator (PRBS), plus a XOR gate
- x^15 + x^14 + 1 polynomial
- Coefficients stored in Flip-Flops



- Data Coding Functions
  - Reed-Solomon encoder
    - Part of Forward Error Correcting (FEC) coding
    - Adds redundant bits
    - Detect and correct errors
    - IP + glue logic

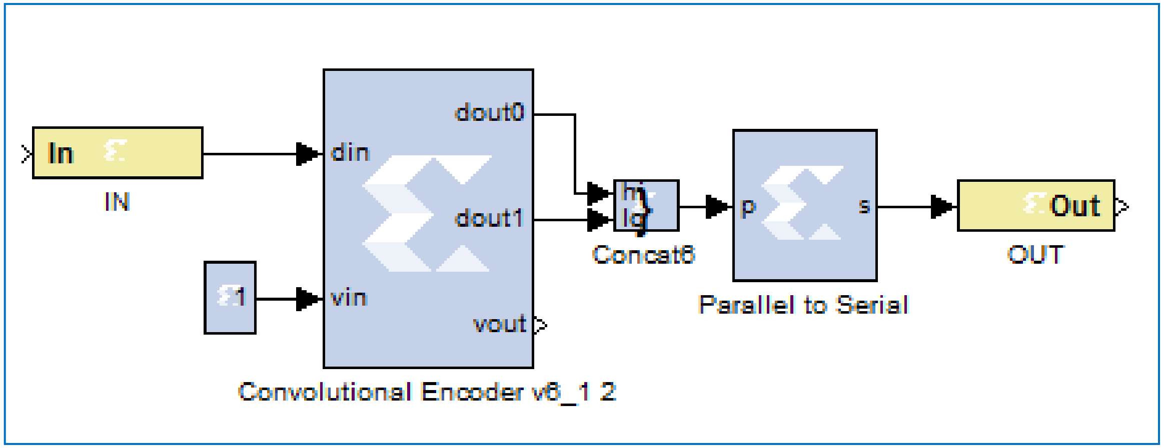


IMPLEMENTATION

Data Coding Functions

– Convolutional encoder

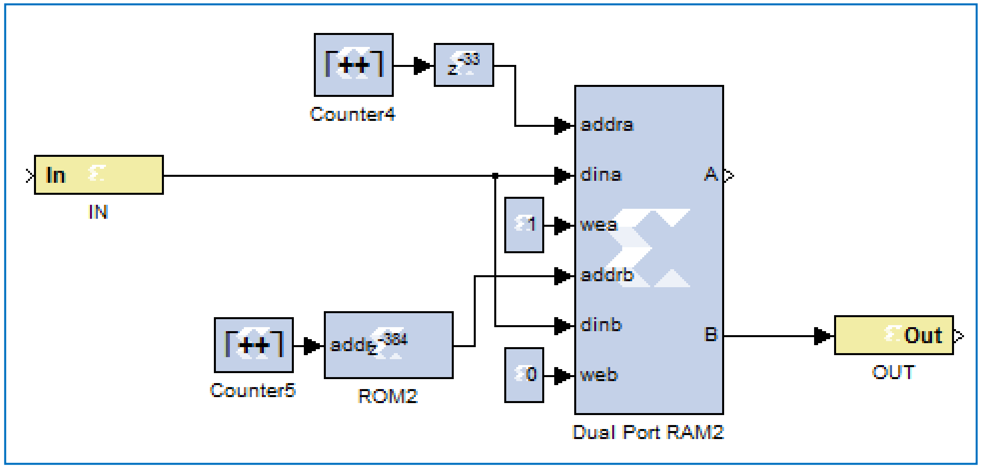
- Part of Forward Error Correcting (FEC) coding
- IP + glue logic



Data Coding Functions

– Puncturer

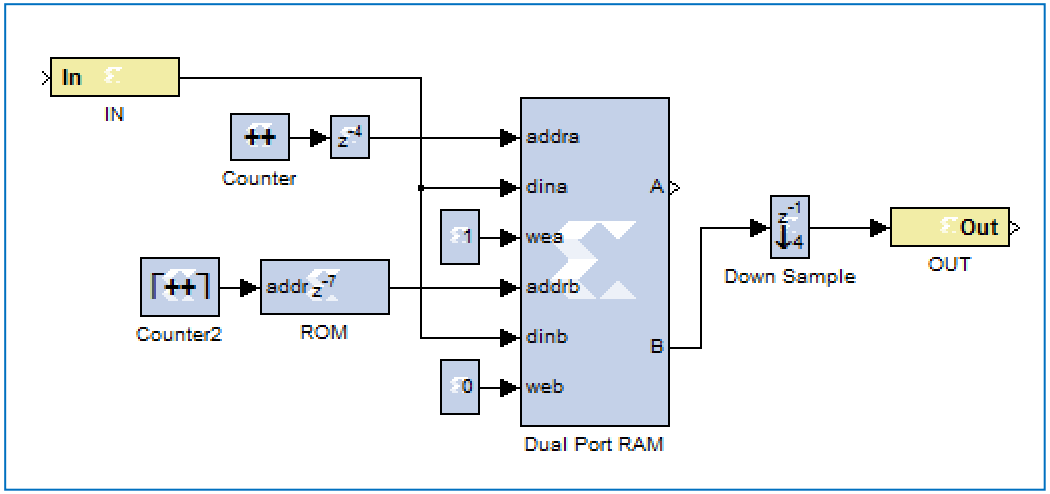
- Some bits selectively deleted
- Usually after a convolutional encoder
- Higher code rates. More flexibility
- Dual port RAM + glue logic



Data Coding Functions

– Interleaver

- Temporal diversity against burst errors
- Ease FEC decoding
- Dual port RAM + glue logic
- Interleaving pattern stored in BRAM

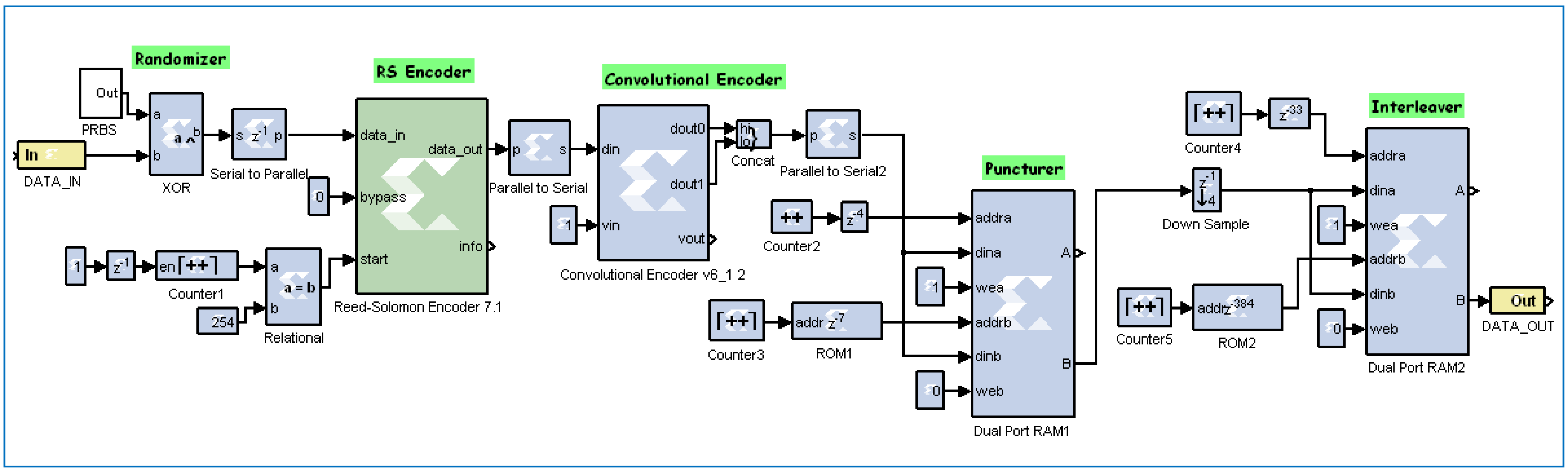


IMPLEMENTATION

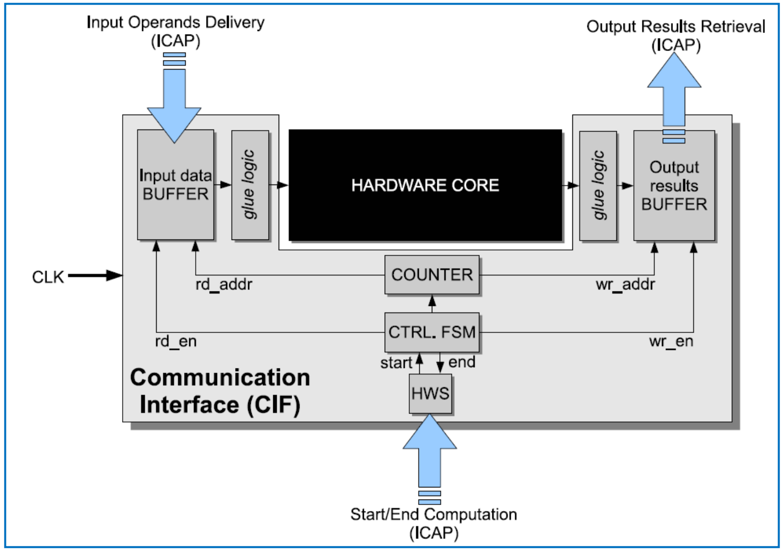
Data Coding Functions

– Full-standard implementation

- Coarse grained execution
- Simulates a complete communication standard change



- I2CI: ICAP-based Inter-task Communication Infrastructure
  - No “wired” communication system
    - Higher area performance
    - More freedom for the allocator
  - Direct data access via ICAP
    - Hardware Semaphore (LUT)
    - BRAM communication
    - System Generator link
      - Clock in, clock enable, clock enable clear
  - Snake strategy
    - Reuse out data BRAM for next task



IMPLEMENTATION

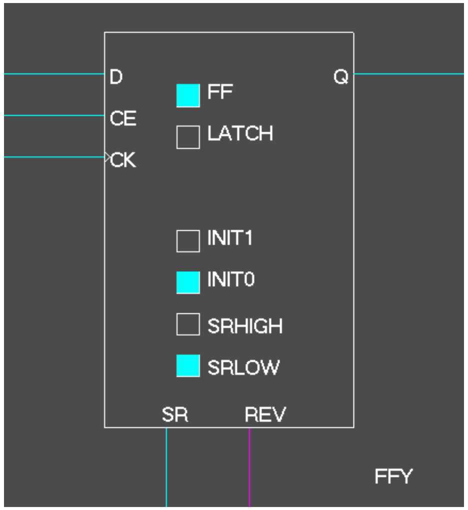
- Function context saving and restoration procedure
  - Preemptive operating system
  - Task context needs to be saved/restored
  - Issues related with Flip-Flops discovered
    - FF are not initialized after a partial reconfiguration
    - FF readback gets the initial values by default
  - Context saving
    - GCAPTURE command (ICAP)
    - Updates the INIT/VALUE bits to the current state of the FF
    - Applied to the whole FPGA (caution!)

Function context saving and restoration procedure

– Context restoration

- GRESTORE command (ICAP)
  - Applied to the whole FPGA -> Not valid
  - Working on a protection technique

- Local reset
  - SR pin
  - Initializes FF with the values in SRMODE bit
  - Manual VHDL coding to grant access to SR
  - Connected to the HW semaphore to automate the procedure



Function Parameterization strategy

- Aims to reduce reconfiguration time (preserve tasks)
- Similar but not equal functions used
  - Different generation polynomial (conv. encoder)
  - Different interleaving pattern
  - Different data sizes
- With a correct task design
  - There is only ONE compatible function
  - Minor resource reconfiguration is carried-out
  - Small overhead may appear
- Examples
  - LUT update (1 frame) in convolutional encoder
  - BRAM update (64 frame) in interleaver

Results

– Resource utilization

	Resource utilization			
Function	SLICEs	LUTs	FLIP FLOPs	BRAMs
Randomization	58	71	45	2
RS encoder	176	311	202	2
Conv. Encoder	80	74	55	2
Puncturing	39	47	41	4
Interleaver	483	503	425	4
Full-standard	798	831	724	6
R3TOS (full)	5571	7383	4157	16
R3TOS (stand-alone)	1793	2778	1157	6

- BRAM use
- Glue logic effect in the individual functions
- R3TOS overhead

Results

– Task configuration time comparison

	Configuration time	
Function	Parameterized version	Normal version
Conv. Encoder	0,855 us	75 us
Interleaver	116 us	413 us

- Convolutional encoder: 1 LUT update
  - 98% reduction with parameterization
- Interleaver: 1 BRAM update
  - 70% reduction with parameterization

Results

– Task execution times

Implementation	Execution time of a single iteration				
	Configuration	Data feeding	Processing	Data recovery	Total
Randomization	75 us	58 us	164 us	58 us	355 us
RS encoder	163 us	58 us	164 us	58 us	443 us
Conv. Encoder	75 us	58 us	164 us	58 us	355 us
Puncturing	42 us	58 us	164 us	58 us	322 us
Interleaver	413 us	58 us	168 us	58 us	697 us
Total funct. by funct.	768 us	348 us	824 us	348 us	2,32 ms
Total funct. by funct. (Snake)	768 us	116 us	824 us	116 us	1,82 ms
Full-standard design	673 us	116 us	227 us	116 us	1,13 ms

- Full-standard execution obtains the best time (fully parallel)
- The snake strategy reduces the execution time a 20%
- Execution of one task at a time (time for 2 tasks at a time = 1,63 ms)
- Smaller task ease allocator's work

CONCLUSIONS AND FUTURE WORK

- Conclusions
  - R3TOS is suitable for SDR implementations
  - The SDR features takes care of a secure communication while R3TOS guarantees efficient and reliable hardware utilization
  - Importance of parameterization
- Future work
  - Improve the state saving and restoration procedure
  - Implement the remaining functions that make up a whole SDR system
  - Develop a design methodology that chooses the optimum task size

QUESTIONS