Symposium on Industrial Embedded System (SIES 2007)

Design of a FFT/IFFT module as an IP core suitable for embedded systems

Julián Viejo, Alejandro Millán, Manuel J. Bellido, Jorge Juan, Paulino Ruiz de Clavijo, David Guerrero, Enrique Ostúa, and Alejandro Muñoz.

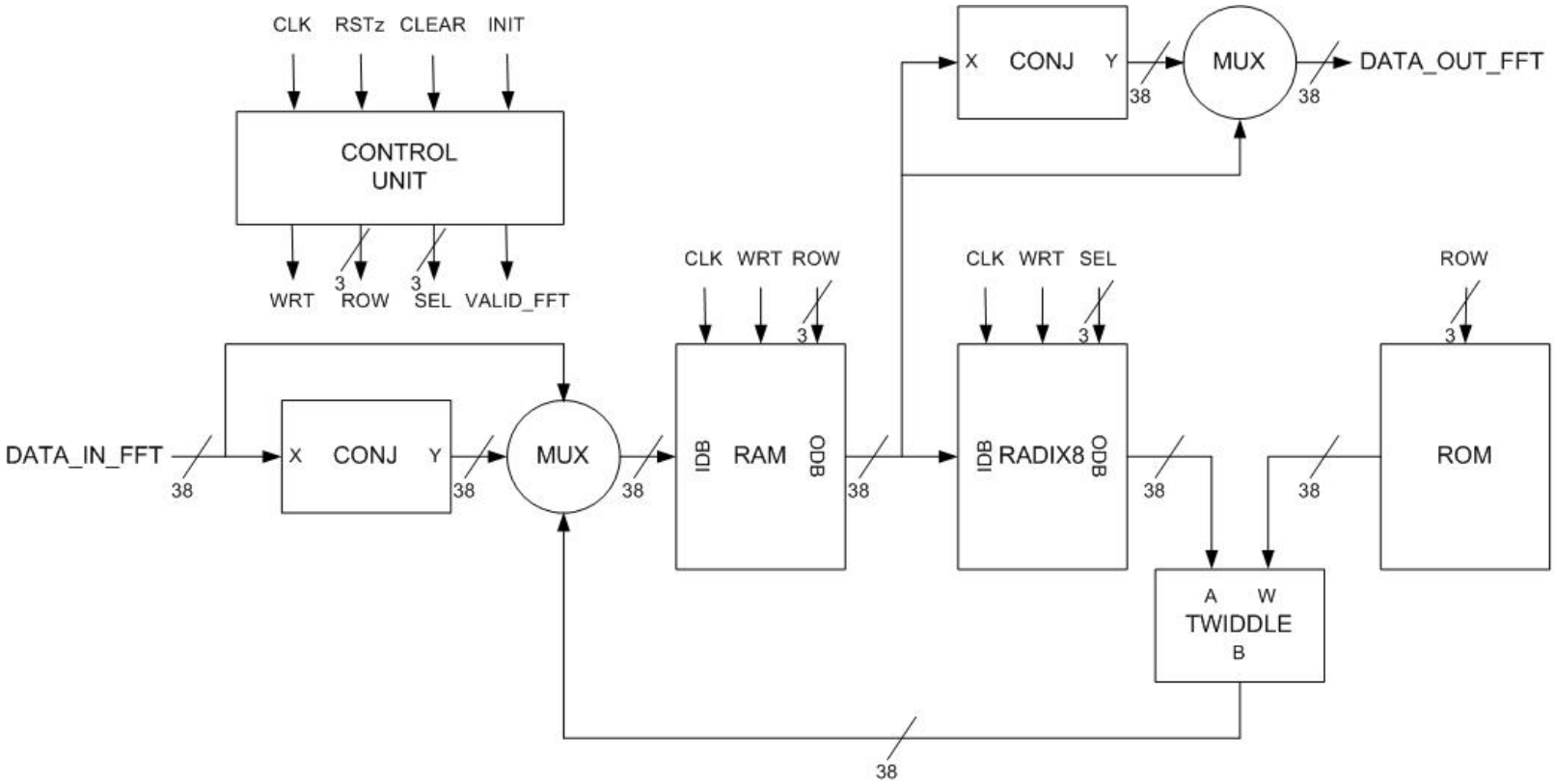
Departamento de Tecnología Electrónica - Universidad de Sevilla (Spain).
Instituto de Microelectrónica de Sevilla - CNM - CSIC (Spain).

Introduction

- Objectives:
  - Perform a comparison between different methodologies:
    - VHDL coding (VC)
    - System-level tools at RT level (STR)
    - System-level tools at macroblock level (STM)
  - Propose a general methodology in order to design this type of DSP functions as an IP core for embedded systems

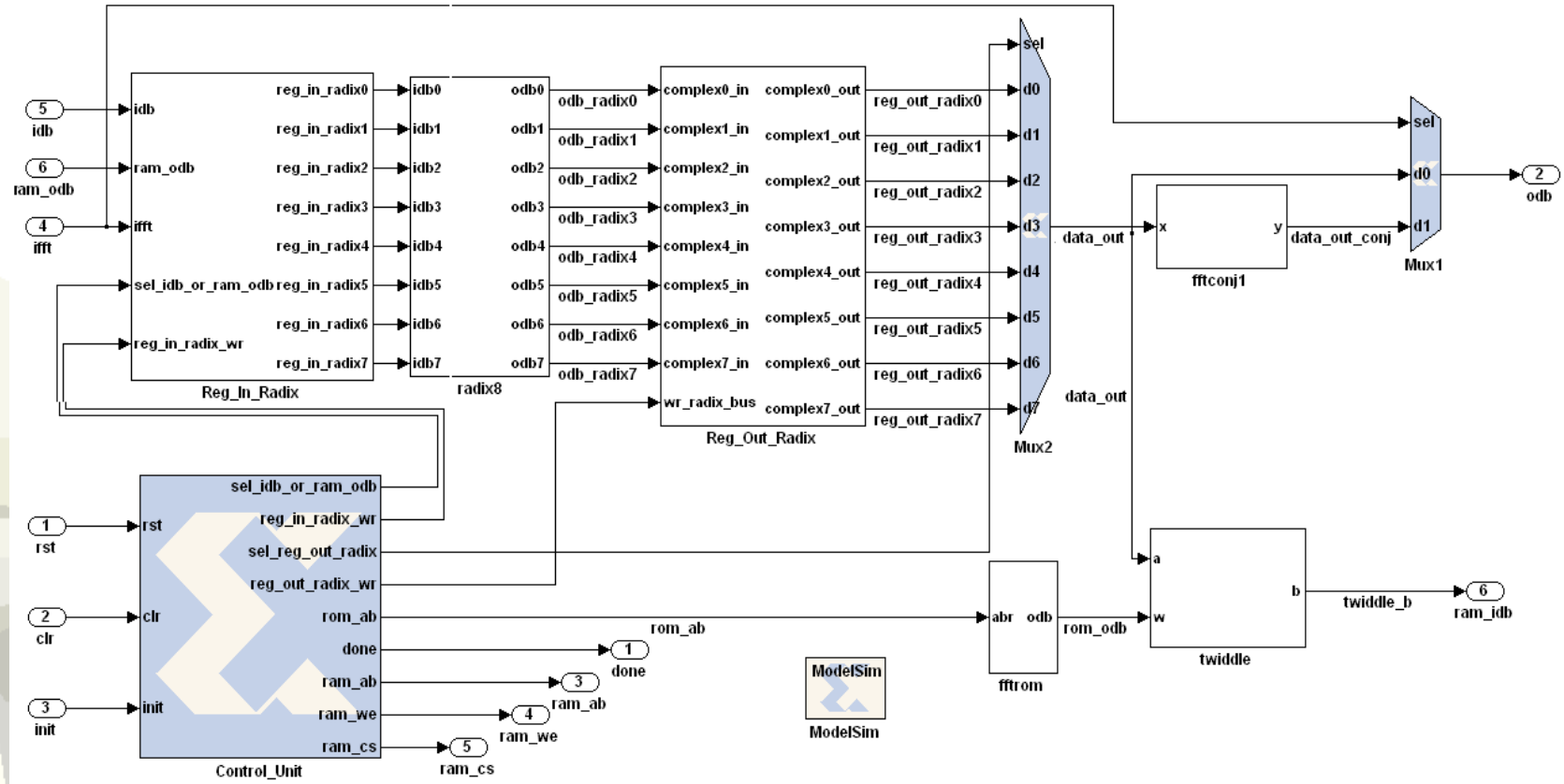
Design methodologies

- VHDL coding (M1):
  - System architecture is designed at RT level
  - Implemented by direct coding in VHDL



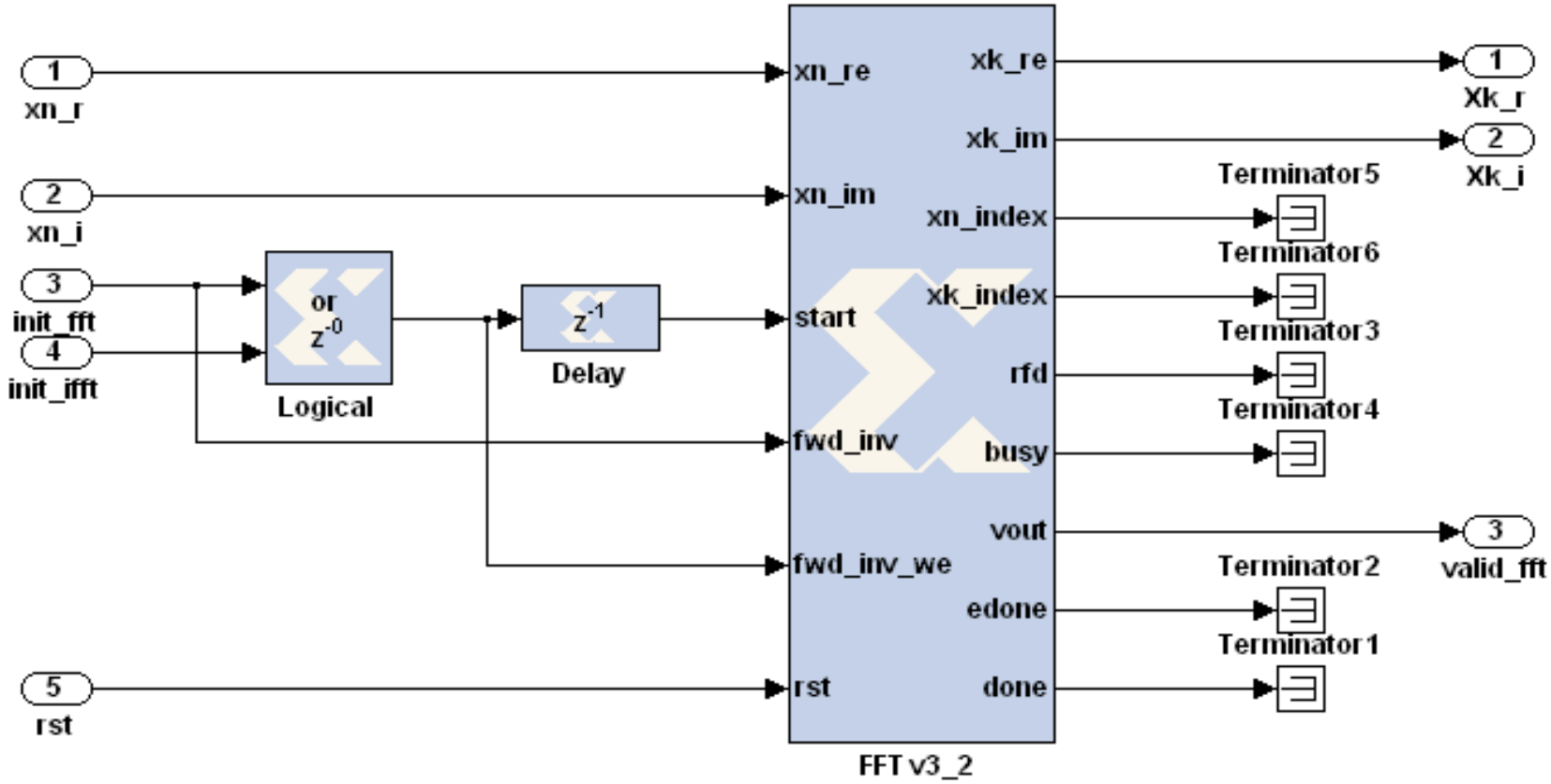
Design methodologies

- System-level tools at RT level (M2)
  - System architecture is designed at RT level
  - Implemented using system-level tools (System Generator for DSP)



Design methodologies

- System-level tools at macroblock level (M3)
  - Module is implemented using the FFT macroblock provided by System Generator



Design results: Simulation

- M1 and M2 methodologies:
  - Designs have been simulated using Simulink and ModelSim (HDL co-simulation)
- M3 methodology:
  - Design simulation is totally carried out using Simulink
- Input stimuli have been generated using the Source Blockset of Simulink

Design results: Simulation

- Simulation results:

	M1	M2	M3
Clock cycles	292	292	341
Mean error	0.59%	0.59%	5.9%

- Discussion:
  - M1 and M2 reduce the amount of clock cycles necessary to the calculation from 341 to 292 with respect to M3
  - M1 and M2 reduce the output relative error from 5.09% to 0.59% with respect to M3

Design results: Hardware implementation

- First approach: designs have been implemented on a Virtex XCV1000 FPGA
- Results obtained for the first approach

	M1	M2	M3
Slices	2056 (16%)	1896 (15%)	1125 (9%)
Slice Flip-Flops	629 (2%)	656 (2%)	1771 (7%)
4 input LUTs	3405 (13%)	3403 (13%)	1789 (7%)
Bonded IOBs	57 (11%)	57 (11%)	57 (11%)
Block RAMs	2 (6%)	2 (6%)	2 (6%)
MULT18x18	-	-	-
GCLKs	1 (25%)	1 (25%)	1 (25%)
Max. operation freq.	26.91 Mhz	25.02 Mhz	62.14 Mhz

- Discussion:
  - M3 saves about 6-7% slices with respect to M1 and M2
  - M3 achieves the highest maximum operation frequency

Design results: Hardware implementation

- Second approach: designs have been implemented on a Virtex-II XC2V2000 FPGA
- Results obtained for the second approach:

	M1	M2	M3
Slices	1214 (11%)	1276 (11%)	-
Slice Flip-Flops	619 (2%)	656 (3%)	-
4 input LUTs	1972 (9%)	2197 (10%)	-
Bonded IOBs	58 (14%)	58 (14%)	-
Block RAMs	2 (3%)	2 (3%)	-
MULT18x18	8 (14%)	4 (7%)	-
GCLKs	1 (6%)	1 (6%)	-
Max. operation freq.	40.02 Mhz	40.03 Mhz	-

- Discussion:
  - Results are very similar for both M1 and M2
  - Designs reach the same maximum operation frequency (40 MHz)

Proposal of a general methodology

- Our proposal combines both M1 and M2 and can be summarized as follows:
  - Design the module with System Generator at RT level
  - Use VHDL coding for those blocks that are easier to program with such a language
  - Verify the module by HDL co-simulation
- This methodology:
  - Facilitates the design and simulation processes
  - Allows designers to maintain total control over the module architecture

Future lines of work

- Improve the module in order to make it fully configurable in terms of data width, symbol length, internal data precision, etc.
- Optimize the design, reducing the hardware resources used and increasing the operation frequency
- Add an interface that allows its connection to a standard bus: OPB, APB, etc.