Table 2 - Reconfiguration time for static and partial reconfiguration on a Xilinx Virtex-2 PRO (XC2VP30)

design type (Static/Partial)	# of slices	Btstrm size (Kbits)	prgrm. time JTAG(ms)	program. time SelectMAP(ms)
static conf	13696	1415	2318	48
DCT8 prtl recnf	378	216	354	7.3
FFT8 prtl recnf	512	426	698	14.3

and one imaginary component. *FFT16's* input timing is different in the way that *start* and *ce* (clock enable) have certain cycle-level specifications described in the previous section. The generated interface meets all those timing requirements. The *FFT16* core's overflow output pin, *OVFLO*, is duplicated and exported by the wrapper to the outside data-path for further use. In *RS_encode*'s output, the first 13 data elements are the data symbols that were fed into the IP. From the point of view of the outside data-path, these data are known and do not necessarily need to be recovered from the IP core again, and only the two check symbols, which follow the first 13 data elements, are needed. The *RS_encode* IP core utilizes output signal *info* to indicate the present of the check symbols. The generated wrapper monitors *info*'s de-assertion and latches the check symbols in an appropriate timing.

ROCCC wraps these IPs so that they have unified outside interface. These four examples illustrate ROCCC's capability to meet various timing protocols of IP cores. The execution time overhead at both the input side and output side for these four examples is one clock cycle. The area of wrappers accounts for 2% ~ 64% of the corresponding wrapped cores. Most of the wrappers' area cost comes from the registers used to do serial to parallel and parallel to serial conversion. Compared to modern FPGAs' capacity, this overhead is quite small.

We measured the time required to load a static bitstream as well as the time required for programming partial bitstreams on the FPGA [Table 2]. JTAG and SelectMAP are two interfaces for reconfiguration of the FPGA. Since the partial bitstreams are smaller in size than the static bitstreams, a partial reconfiguration can be achieved in a shorter time vis-à-vis complete reconfiguration.

7. CONCLUSION

Increasing silicon capacity requires both higher level design methods and easier intellectual property core reuse. ROCCC, a reconfigurable computing compiler, is designed to take applications in C as input and generate RTL VHDL code. In this paper, we introduced one aspect of ROCCC's functionalities, the IP wrapper generation.

As the input to the ROCCC system, users write IP wrappers in high-level timed C. Clock cycle delays are described as function calls and users do not have to implement any cycle-level details in the input abstraction. Constrained by the delay function calls, ROCCC converts the wrapper from control flow graph to data flow graph. The compiler schedules pipelined instructions using predication. Wrapped IP cores have identical interface compared with the outer predicated circuit that also generated by ROCCC.

The wrappers of the IP core examples meet the various timing protocol requirements, and unify the IP cores' interface with the outer compiler-generated circuit. The results show that the execution time and area overhead are reasonable low. We also show the same tool can be used to support run-time reconfiguration on FPGAs by generating one wrapper that interfaces to multiple cores.

8. REFERENCES

[1] SUIF Compiler System. http://suif.stanford.edu, 2006

[2] Machine-SUIF. 2006 http://www.eecs.harvard.edu/hube/research/machsuif.html

[3] G. Holloway. The Machine-SUIF Static Single Assignment Library. Division of Engineering and Applied Sciences, Harvard University 2002.

[4] Z. Guo, B. Buyukkurt, W. Najjar and K. Vissers. Optimized Generation of Data-path from C Codes for FPGAs, Int. ACM/IEEE Design, Automation and Test in Europe Conference (DATE 2005). Munich, Germany, March, 2005.

[5] Virtual Socket Interface Association (VSIA), http://www.vsi.org/ , 2006

[6] P.Chou, G. Ortega, G. Borriello. Interface co-synthesis techniques for embedded systems, Int. Conf. on Computer Aided Design, San Jose, USA,1995.

[7] R. Lysecky and F. Vahid. Prefetching for Improved Bus Wrapper Performance in Cores, ACM Transactions on Design Automation of Electronic Systems, Vol. 7, No. 1, pp. 58-90, January 2002.

[8] J. Zhu. MetaRTL: Raising the abstraction level of RTL design, Design, Automation, and Test in Europe, Munich, Germany, 2001

[9] V. Stuikys, R. Damasevicius. Soft IP Customisation Model Based on Metaprogramming Techniques, Informatica, Lith. Acad. Sci. 15(1): 111-126 (2004)

[10] J. Tripp, K. Peterson, C. Ahrens, J. Poznanovic, M. Gokhale. Trident: An FPGA Compiler Framework for Floating-Point Algorithms, int. Conference on Field Programmable Logic and Applications (FPL 2005). Finland, 2005.

[11] R. Mukherjee, A. Jones, P. Banerjee, System Level Synthesis of Multiple IP Blocks in the Behavioral Symthesis Tool, Int. Conf. on Parallel and Distributed Computing and Systems (PDCS), November 2003.