Field Programmable MCM Systems — Design of an Interconnection Frame

Ivo Dobbelaere[*], Abbas El Gamal[*] Dana How[†]and Bendik Kleveland

Information Systems Laboratory, Stanford University, Stanford, CA 94305

Abstract—A field programmable MCM architecture utilizing an array of modified FPGAs is proposed. The fast interchip interconnections required by large partitioned designs are realized with a fixed wiring network on the MCM and programmable interconnection frames on each FPGA.

1 Overview

We present a new field programmable architecture for prototyping large designs using multiple FPGAs which offers excellent performance and economy while retaining the immediate turnaround of FPGAs at the system level.

For our packaging technology we have selected MCMs[1] because they offer the large pin count per chip necessary for high chip utilization in a partitioned design, and because the off chip delays are smaller than with PCBs. Due to the technological complexities of providing configurability at the MCM level we assume a fixed, nonconfigurable, statistically-determined wiring pattern.

It would be convenient to use commercially available FPGAs (e.g. Actel[4] and Xilinx[5]), but these are not suited to our requirements. First, such FPGAs do not provide the high pin to gate ratio required when partitioning a design among multiple FPGAs. As a result the FPGAs are typically underutilized[6]. Secondly, since the I/O buffers of these chips are designed for general purpose use with PCBs, they do not give the better performance possible with MCMs.

Responding to these two considerations, we could use an FPGA with a commercial internal architecture, yet with a high pin count and special I/O buffers. However, the fixed MCM wiring pattern assumed in our architecture imposes severe routing constraints which we solve by routing many signals through the FPGAs themselves[6]. Since the routing architectures of commercial FPGAs are optimized for local interconnects, the delays incurred in routing signals from one pin to another through the entire chip are too large. Placing specialized switching chips[9] on the MCM would partially avoid this problem, but such centralization would increase average wirelength and lead to inflexibility in partitioning, placement and routing.

Therefore, we propose to modify the FPGAs to support quick connections from one pin to another, thereby uniformly distributing routing resources across the chips on the MCM. We do so by surrounding the FPGA logic core with an *interconnection frame* (Fig. 1) which supports fast thru-chip as well as chip-to-chip connections.

In this paper we focus on the configuration and the design of the interconnection frame. In section 2 we describe the proposed architecture of the interconnection frame. In section 3, we use circuit simulation to determine the circuit types and the transistor sizing assuming AT&T's MCM technology[7] and a standard 1.2-μm CMOS technology.

2 Frame Architecture

We assume the logic core of our modified FPGA to be similar to an existing FPGA (specialized functions could also be used). We also assume that the interconnection frame is programmable via a shift register. Connections leaving the MCM are similar to conventional PCB connections and will not be addressed in this paper. Among inter-chip connections within an MCM, connections between neighboring chips will predominate, so many can be implemented as fixed core-to-core connections. The remainder should be programmable using the interconnection frame.

The frame supports two basic switching patterns:

- Connect a signal from the core of chip A through a pin on chip A, across the MCM and to a pin on chip B. The signal is switched through B's frame and then connects to B's core (Fig. 2a). This is the conventional pattern.
- Connect a signal from the core of chip A through

[*]Partially supported by FBI contract J-FBI-89-101

[†]Partially supported by an NSF Graduate Fellowship

0-7803-0246-X/92 $3.00 © 1992 IEEE

a pin on chip A, across the MCM and to a pin on chip B. The signal is switched through B's frame and then connects to another pin on B. The signal is then sent from this pin over the MCM to a pin on chip C which connects to C's core (Fig. 2b). In section 3 we optimize this pattern, since it is more critical than the previous pattern.

The frame is split between the four corner regions of the chip, each containing a switchmatrix (Fig. 3a). Within a switchmatrix, all horizontal lines can connect to all vertical lines (Fig. 3b). Connections between the four switchmatrices and between the switchmatrix and the chip core are also provided.

With MCMs and flip-chip bonding, the direct interconnections between neighboring chips have a much lower capacitance than interconnections between remote chips. Hence, in considering thru-chip interconnection patterns, we will distinguish short interconnections between neighboring chips (5mm and about 1pF for AT&T), and long interconnections between chips that are not close (anything over 2cm); intermediate lengths could be classified either way (Fig. 1). Our goal is to take advantage of the performance benefits of the MCM environment by optimizing the switchmatrix for the expected mix of attached interconnections. Depending on the length of the connections to the switchmatrix, we have the four thru-chip interconnection cases optimized in section 3: short-to-short, long-to-long, short-to-long, or long-to-short.

At the inputs to the switchmatrix, small buffers may be present to enhance the incoming signal. NMOS transistors, whose gates are driven by 7V to eliminate the voltage drop problem, route the 5V signals inside the switchmatrix. Additionally, output buffers may be needed to drive the off-chip interconnect. Since every signal path must be bidirectional, any output buffer must have an input buffer placed back-to-back; or the bidirectionality may be obtained by simply placing a pair of large programmable bypass transistors across the output buffer (Fig. 3c).

3 Frame Design

3.1 MCM Model

Throughout this section we assume parameters for AT&T's MCM technology[8], with a wiring capacitance of about 1pF/cm, a characteristic impedance between 50 and 100Ω, and a wire resistance of 20mΩ/□. We chose 50μm×50μm flip-chip bonding pads, with a capacitance of 0.5pF and an inductance of 0.05nH.

3.2 Buffer Circuits

In our application differential lines are unacceptable because of the higher pin count. The advantage of differential lines is mainly noise immunity; the gain in speed is not compelling. In CMOS, there is a choice between low swing ECL-like circuits[2], low swing CMOS[3], and standard full swing CMOS. The low swing ECL-like circuits are not interesting for our application because of the high DC power consumption. Low swing CMOS circuits consume less power, but simulations show them to be slower, mainly due to the low swing to high swing conversion at the receiver end. Our final choice is full swing CMOS buffers.

3.3 Path Design

We simulated each of the four interconnection cases with HSPICE — Fig. 4 summarizes the signal path we simulated in the short-to-short case; the others were similar. The input buffer is a two stage tristate buffer with adjustable sizing. The output buffer is a three stage tristate buffer with sizing such that MCM transmission line behavior is insignificant (the driver sizes are W_p=400 μm, W_n=200 μm). For each case, we considered the placement of either an input bypass transistor pair or a buffer at the switchmatrix input, and the placement of a buffer at the matrix output, for switchmatrices varying in transistor widths and in number of short and long paths connected. The propagation delay and the rise and fall times of the output signals were measured for T=70°C. We have measured rise times to the 80% point since 7V NMOS switches degrade the rising edge past 4V, which is not critical for our circuits.

The first simulations indicate that input buffers do not improve the delay for *any* of the interconnection cases. In fact, for different interconnection lengths, input buffers added 0.5ns to the delay even for the fastest input buffers, so we will not use them. However, each path must support signal propagation in both directions, so where there is an output buffer (as determined below) we will place an input bypass transistor pair. Fortunately, a large input bypass transistor pair will not significantly increase the total propagation delay.

Similarly, the best delay with no output buffer in the short-to-short case is at least 20% smaller than the best delay with a buffer, whether or not the signal previously passed through an input bypass transistor pair. The risetime is longer without an output buffer, but not unacceptably so (in the range of 2.5ns). No output buffer should be used in the short-to-short case.

In the case of a long-to-long interconnection, the presence of an output buffer does not significantly affect the best delays. However, since the output buffer does improve the risetime it should be included. In addition, the presence of an output buffer makes small delay times possible without large switch transistor

Case	Input	Output
short-to-short	no bypass	unbuffered
short-to-long	no bypass	buffered
long-to-short	bypass transistors	unbuffered
long-to-long	bypass transistors	buffered

Table 1: Optimized Path Configurations

sizes. As previously argued, with the output buffer we must include a large input bypass transistor pair for bidirectionality.

The considerations for the short-to-long case are very similar to the long-to-long case above. Consequently the short-to-long case will not use an input buffer but it will use an output buffer.

The long-to-short case is the reverse of short-to-long, so there must be an input bypass transistor pair. Omitting the output buffer does decrease the delay (by 10%), but this would increase the risetime by 3ns. However, we have decided that for consistency with the implementations already presented for the other three cases, the increased risetime is acceptable (since the switching delay from propagation and risetime is still less than a long-to-long interconnection delay), and so we will not use an output buffer in the long-to-short case.

Table 1 summarizes the path configurations for each case which provide the most effective system solution. From all the arguments we see that we should place an input bypass transistor pair and output buffer combination at a pin if and only if it is attached to a long interconnection. This is the simplest rule which will effectively implement all four thru-chip interconnection cases.

3.4 Transistor Sizing

From the equivalent circuits for the path configurations selected above, we obtain the first-order-delay-minimizing switch transistor sizes W_{ss}, W_{sl}, W_{ls}, and W_{ll} for the four cases:

$$W_{ss} = \sqrt{\frac{C_{x,s} + N_l W_{sl} c_i}{2 N_s R_o g_t c_i}} \qquad (1)$$

$$W_{sl} = \sqrt{\frac{N_l W_{ll} c_i}{(N_l + N_s) R_o g_t c_i}} \qquad (2)$$

$$W_{ls} = \sqrt{\frac{C_{x,s} + N_s W_{ss} c_i}{(N_s + N_l)(R_o + R_i) g_t c_i}} \qquad (3)$$

$$W_{ll} = \sqrt{\frac{N_s W_{ls} c_i}{2 N_l (R_o + R_i) g_t c_i}} \qquad (4)$$

N_s and N_l are the number of short and long interconnections, respectively, attached to the switchmatrix; R_i is the input bypass transistor pair ON resistance; R_o is the output driver transistor equivalent resistance; $C_{x,s}$ is the external short interconnect capacitance; g_t is the switch transistor ON conductance per μm transistor width; and c_i is the sum of the switch transistor drain-to-source capacitance and the metal wire capacitance, per μm transistor width. Note that the widths derived above for minimal first order delays depend on the switchmatrix sizes N_s and N_l.

Simulations show that the delay dependence on the widths in the short-to-long and long-to-long cases is small because the buffer shields the switchmatrix from the capacitance of the external interconnection. Therefore, we are not constrained to the values calculated for W_{sl} and W_{ll} in (2) and (4), but have considerable flexibility in their selection if we maintain a certain minimum width (15μm for the chosen parameters).

However, the paths without an output buffer (short-to-short and long-to-short) show a much greater delay dependence on width; so we should use the values calculated for W_{ss} and W_{ls} in (1) and (3). Since long-to-short is short-to-long reversed, and we showed that the choice of W_{sl} is quite flexible, we will set W_{sl} equal to the value obtained for W_{ls} from (3). This is greater than the minimum width requirement for W_{sl}.

Lastly, we showed that the choice of W_{ll} is unconstrained. Therefore we can choose W_{ll} to save area, as long as we maintain the minimum width.

4 Conclusion

We have investigated alternative implementations of an interconnection frame for FPGAs used on an MCM substrate with fixed wiring. We determined that conventional CMOS full swing signals would afford the best inter-chip performance rather than some form of low swing signalling. Surprisingly, I/O buffers were not needed in many cases due to the decreased MCM capacitance and switches controlled by 7V signals.

Fig. 5 shows simulated delay versus switchmatrix dimensions of our chosen implementation for the short-to-short case. Although we have yet to determine what mix of short and long connections our prototyping architecture will require, this graph and similar ones for the other cases show that neighboring connection delays of 3.5ns and long-range connection delays of 6ns are attainable when routing a signal through a chip with the interconnection frame. Such speeds will make our prototyping architecture feasible.

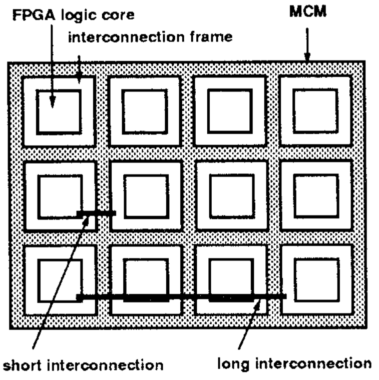

Figure 1: Multiple FPGAs placed on an MCM.

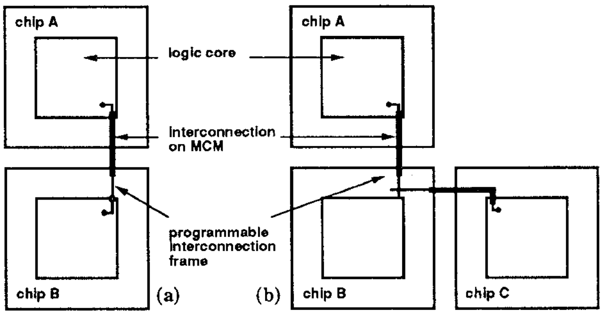

Figure 2: The frame supports two switching patterns.

References

[1] R. Tummala and E. Rymaszewski, *Microelectronics Packaging Handbook*, Van Nostrand Reinhold, 1989.

[2] B. Chappell *et al*, "Fast CMOS ECL Receivers with 100-mV Worst-Case Sensitivity," *IEEE J. Solid State Circuits*, 23-1, February 1988, pp. 59-67.

[3] T. Knight, Jr. and A. Krymm, "A Self-Terminating Low-Voltage Swing CMOS Output Driver," *IEEE J. Solid State Circuits*, 23-2, April 1988, pp. 457-464.

[4] A. El Gamal *et al*, "An Architecture for Electrically Configurable Gate Arrays," *CICC* 1988, pp. 15.4.1-4.

[5] H. Hsieh *et al*, "A 9000-Gate User Programmable Gate Array," *CICC* 1988, pp. 15.3.1-7.

[6] S. Walters, "Computer-Aided Prototyping for ASIC-Based Systems," *IEEE Design and Test of Computers*, June 1991, pp. 4-10.

[7] K. Tai, "Si-on-Si MCM Technology and the Initiation of a University MCM Program," *Multichip Module Workshop*, Santa Cruz, CA, March 1991, pp. 10-13.

[8] J. Burr *et al*, "System-wide Energy Optimization in the MCM Environment," *Multichip Module Workshop*, Santa Cruz, CA, March 1991, pp. 66-83.

[9] "Startup Aptix details its FPIC architecture," *Electronic Engineering Times*, January 6, 1992, issue 674.

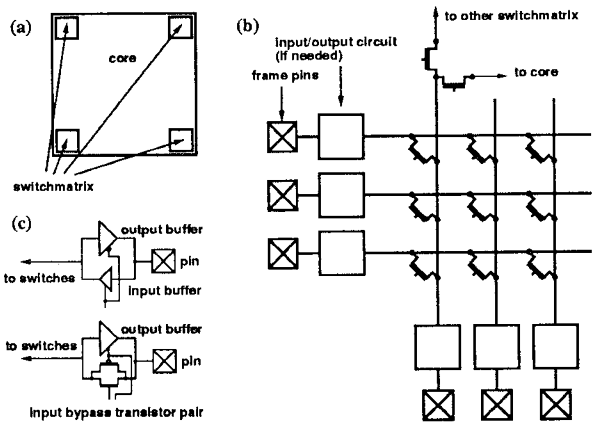

Figure 3: (a) Switchmatrix placement. (b) Switchmatrix structure. (c) I/O circuits.

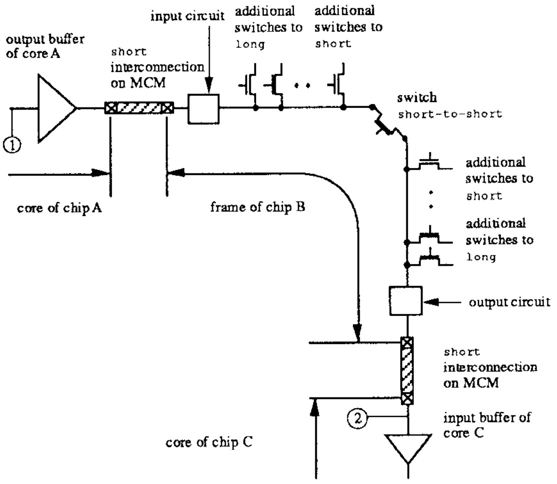
input circuit
additional switches to long
additional switches to short
output buffer of core A
short interconnection on MCM
switch short-to-short
1
core of chip A
frame of chip B
additional switches to short
additional switches to long
output circuit
short interconnection on MCM
2
input buffer of core C
core of chip C

Figure 4: short-to-short signal path from 1 to 2.

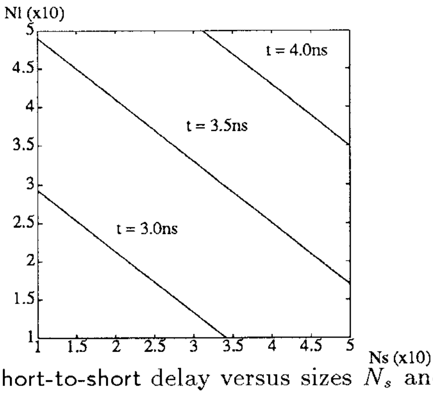

Figure 5: short-to-short delay versus sizes N_s and N_l.