Reducing Cross-Coupling among Interconnect Wires in Deep-Submicron Datapath Design

Joon-Seo Yim

DSP Group, Information Technology Lab.
LG Corporate Institute of Technology
16, Woomyeon-Dong, Seocho-Gu,
Seoul, 137-140, Korea
e-mail: jsyim@lotus.lgcit.com

Chong-Min Kyung

Department of Electrical Engineering
KAIST
373-1, Kusong-Dong, Yusong-Gu,
Taejon, 305-701, Korea
e-mail kyung@ee.kaist.ac.kr

Abstract

As the CMOS technology enters the deep submicron design era, the lateral inter-wire coupling capacitance becomes the dominant part of load capacitance and makes RC delay on the bus structures very data-dependent. Reducing the cross-coupling capacitance is crucial for achieving high-speed as well as lower power operation. In this paper, we propose two interconnect layout design methodologies for minimizing the "*cross-coupling effect*" in the design of full-custom datapath. Firstly, we describe the *control signal ordering* scheme which was shown to minimize the switching power consumption by 10% and wire delay by 15% for a given set of benchmark examples. Secondly, a *track assignment* algorithm based on *evolutionary programming* was used to minimize the cross-coupling capacitance. Experimental results have shown that the chip performance improvement as much as 40% can be obtained using the proposed interconnect schemes in various stages of the datapath layout optimization.

1 Introduction

With technology scaling down into deep submicron(DSM) regime, circuit designers need to deal with a growing number of issues previously considered with the second order priority[1, 2, 3, 4]. Nowhere is this more true than in the design of interconnect. The aggressive pitch and aspect ratio(thickness over width) of today's advanced processes are such that capacitance to neighboring metal wires can be larger than the capacitance to metal above and below. For example, the wire aspect ratio has reached 2.18(= 0.96/0.44) for 0.25μm TLM(Triple Layered Metal) process as shown in Fig. 1. The horizontal capacitance(C_h), already as the main source of the coupling between metal wires, can be several times larger than the vertical capacitance(C_v) as shown in Table. 1.

For higher metal layers(M3), the inter-wire coupling effects are more pronounced due to the increased metal heights and the lessened substrate effects. A scheme for layout method of reducing these cross-coupling effects is to put upper and/or lower ground planes to divert electric field lines to the ground plane rather than the neighboring signal.

This larger horizontal capacitance causes higher cross-coupling between wires and also makes RC delay on bus structures very data-dependent. As a result of the "*Miller effect*", this horizontal capacitance varies depending on the switching behavior of a victim wire and its neighbors. For example, consider two wires that run in parallel for a significant distance with minimum space. If one wire is switching from low to high while its neighbors are simultaneously switching from high to low as shown in Fig. 2(a), the effective capacitance between the wires becomes doubled compared to the case when the neighbors are quiet. On the other hand, if both neighbors are switching in the same direction at the same time as shown in Fig. 2(b), then the horizontal capacitance becomes effectively zero.

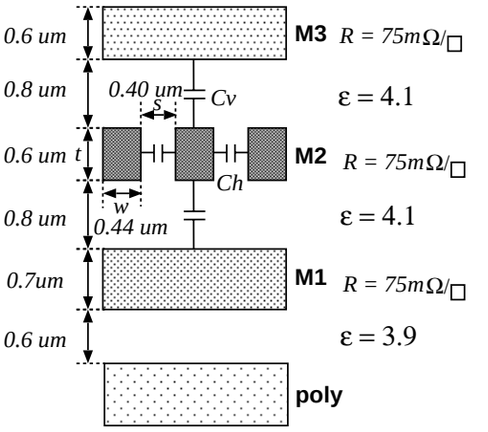

Figure 1: *The crosssection of the metal interconnect in 0.25 μm CMOS TLM process showing the minimum geometries with the oxide thinkness T_{ox} = 6nm. Dielectric constant and sheet resistance are also shown.*

Table 1: *Wire capacitance of 1000μm long metal with* w= 0.44μm *and* s= 0.56μm

		$C_{wire}(C_v, C_h)$	C_v:C_h
without coupling	M1	221 fF (101, 120)	1:1.20
	M2	205 fF (90, 115)	1:1.28
	M3	206 fF (54, 152)	1:2.81
with coupling	M1	341 fF (101, 240)	1:2.38
	M2	319 fF (90, 229)	1:2.54
	M3	358 fF (54, 304)	1:5.63

For a long interconnect wire, where coupling capacitance dominates the total load capacitance, the wire delay can vary by several scores of percent as a function of the switching activities of neighboring wires as shown in Fig. 3.

Note that the switching behavior of wires above and below can also affect delay. However, wires on alternating lay-

DAC 99, New Orleans, Louisiana

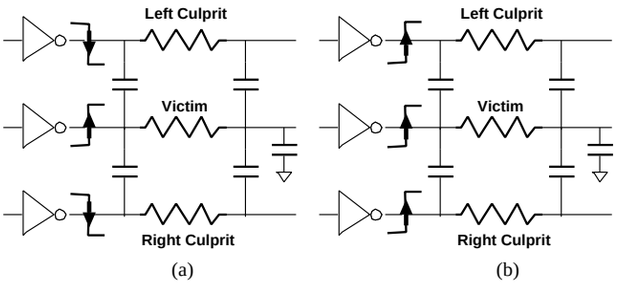

Figure 2: *Cross-coupling effect on the victim wire from the adjacent nets: (a) the worst-case and (b) the best-case.*

ers generally run perpendicular to each other and are not tightly related in functionality, therefore do not have a significant effect on the delay. The most obvious way to reduce this data-dependency is to increase the effective space between the metal wires, which reduces the percentage of horizontal component among total capacitance and, therefore, reduces the data-dependency of the interconnect delay. Another way of mitigating this effect is to carefully order or interleave busses so that simultaneous opposite transitions between neighboring wires are suppressed as much as possible. For example, in a dynamic bus, all wires are pulled up in the precharge phase. Therefore, the coupling capacitance does not slow down the transition. During the evaluation phase, all signals are guaranteed to make monotonic falling transitions, which guarantees that no wire need to fight against two neighbors switching in opposite direction.

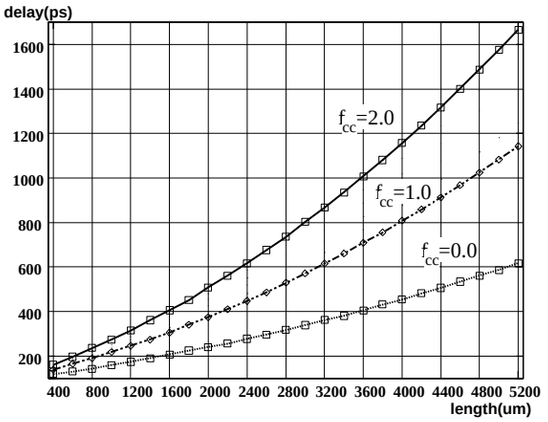

Figure 3: *Dependence of bus interconnect delay vs. wire length for various coupling factors(f_{cc}): $f_{cc} = 2$ implies that neighbors make opposite transitions, $f_{cc} = 0$ implies that they are switching in the same direction, while $f_{cc} = 1$ implies that they are static.*

Various efforts have been exercised to minimize the cross-talk in the routing[5, 6, 7, 8, 9, 10, 11, 12]. Especially. Chaudhary[5] presented a linear programming approach to increase the space between wires, which, in turn, reduces the cross-coupling noise. However, this approach is limited in that it cannot change the ordering between the wire segments. Gao[6] proposed a track permutation technique with the initial gridded channel routing based on linear programming. Jhang[7] proposed a repeated segment rearrangement scheme, where the neighboring nets which are placed at adjacent tracks are moved to another track to minimize the coupling capacitance.

In the high-performance datapath design, which is usually done by full custom layout approach, detailed layout and the interconnect schemes need to be planned before the layout composition. Assuming the functional elements are placed vertically, vertical interconnect is used for the inter-element routing tracks and horizontal interconnect is used for the routing of control signals. For the full custom datapath layout in mind, we propose the *control signal ordering* scheme in the horizontal direction and a *track assignment* algorithm in the vertical direction to minimize the cross-coupling effect considering the switching behavior.

In section 2, we propose an effective *control signal ordering* scheme. Section 3 shows *track assignment* algorithm. Both layout methods minimize the cross-coupling effect. Experimental results based on real microprocessor examples are shown in section 4.

2 Control Signal Ordering Minimizing the Cross-Coupling Effect

In the datapath design, a lot of multiplexers are needed for the RTL functionalities and most of control signals are related to the multiplexers. To implement low power and high-speed multiplexer, pass transistor logic gates[13, 14] are commonly used to reduce the intermediate node transitions as shown in Fig. 4. In the multiplexer design, the control signal timing often becomes very crucial to the chip performance. These control signals are generated from the standard cell block, globally routed to the datapath, and then drive the multi-bit datapath multiplexer cells handling as many as 64 bits. Therefore, the multiplexer selection signals usually lie in the time-critical path. If the control signals are overlapped in time, then multiple transistors are activated simultaneously. This leads to the flow of excessive short-circuit current, which results in the delay of data transfer from the input port to the output port.

Interestingly, one pair of selection signals, *i.e.*, `s` and `sb`, for each pass transistor makes a transition in the opposite direction as shown in Fig. 4. Accordingly, these are very vulnerable to be damaged by the cross-coupling effect. In this section, we propose a *"control signal ordering scheme"* minimizing the cross-coupling effect between the control signals.

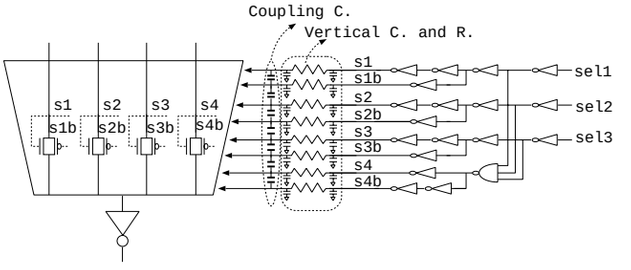

Figure 4: *A 64-bit multiplexer example showing the control signal cross-coupling effect.* `si` *and* `sib` *are complementary each other.*

A conventional schemes for the ordering of multiplexer selection signals is shown in Fig. 5, which is very sensitive to the cross-coupling effect. For example, assuming that the selection is changed from 1 to 2. Then, there are opposite transitions in `s1/s1b` pair and `s2/s2b` pair in the "*order1*" layout. The situation is not improved with the "*order2*"

layout, which allows opposite transitions in the `s1/s2` pair and `s1b/s2b` pair. All the transition cases for the multiplexer selections are shown in the right side of Fig. 5. For N-input multiplexer, the notation t_{ij}(`i,j`=*1,...N*) implies that there is a switching of the selection signal from `i` to `j`. t_{ii} does not invoke the signal transition, while t_{ij}(`i` $\neq$ `j`) leads to the signal transition which is related to the multiplexer delay and the power consumption. In the layout point of view, "*order2*" is preferred to "*order1*" as shown in Fig. 6.

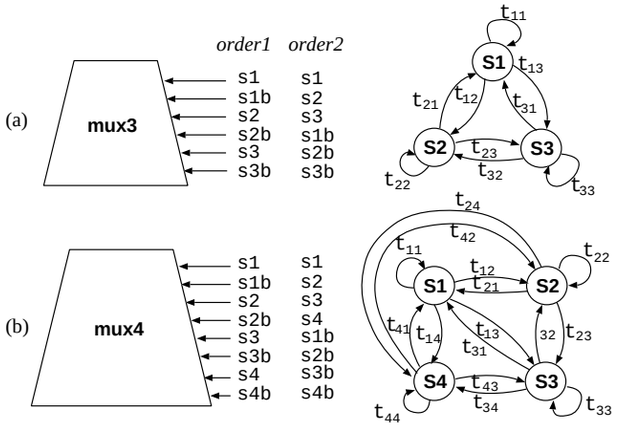

Figure 5: *Two conventional control signal orderings(order1, order2) and transition for (a) 3-to-1 mux (b) 4-to-1 mux*

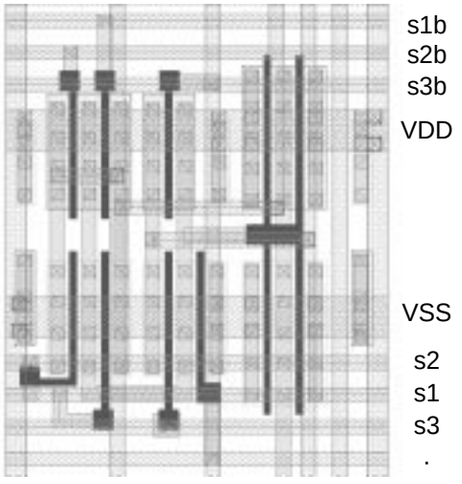

Figure 6: *Multiplexer layout with "order2" scheme where the selection signals are drawn horizontally using metal 2.* `s1,s2,` *and* `s3` *are connected to NMOS, while* `s1b,s2b,` *and* `s3b` *are connected to PMOS.*

There are two observations related to the multiplexer selection signals.

- observation 1 : `si/sib` signal pair always make transitions in the opposite direction.
- observation 2 : For the change of selection from `i` to `j`, there are opposite transitions between `si/sj` pair and `sib/sjb` pair.

To eliminate the signal switching in the opposite direction, we can construct the following rules.

- rule 1 : Do not place `si/sib` signal pair in neighborhood.
- rule 2 : Do not place `si/sj` signal pair in neighborhood for all `i,j`.
- rule 3 : Do not place `sib/sjb` signal pair in neighborhood for all `i,j`.

In conclusion, only `si/sjb`(`i` $\neq$ `j`) signal pair can be placed in neighborhood to avoid switching in the opposite direction, which is harmful in terms of not only noise coupling but also switching power consumption. Fig. 7 shows the ordering of selection control signals for the 3-input and 4-input multiplexers, respectively satisfying the rules above. These multiplexer selection signal ordering scheme can be generalized onto other control signals, such as barrel shifter. This will be examined as future works.

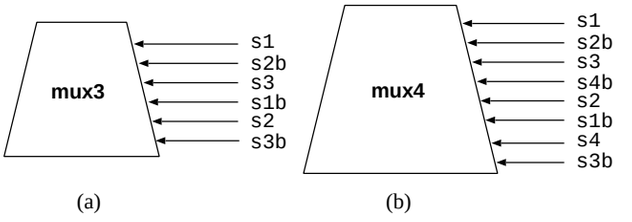

Figure 7: *Proposed ordering scheme of selection control signals minimizing the cross-coupling effect for (a) 3-input and (b) 4-input multiplexers.*

Fig. 8 summarizes the comparison of the frequency of opposite transitions among three different orderings for the 3-to-1 input multiplexer; there are twelve opposite transitions in the "*order1*", eight opposite transitions in the "*order2*", and zero opposite transitions in the "*proposed*" ordering. The situations for the 4-to-1 multiplexer are shown in Fig. 9.

	Selection signal changes from a to b (a->b)						total number of opposite transitions
	1->2	1->3	2->1	2->3	3->1	3->2	
(a)	s1 s1b s2 s2b s3 s3b	 0 1 	 0 1	0 1 	 0 1 	0 1 	12
(b)	s1 s2 s3 s1b s2b s3b	 0 1 	 0 1	0 0 1 	 0 1 	0 1 	8
(c)	s1 s2b s3 s1b s2 s3b	 1 0 	 0 1	0 1 	 1 0 	0 1 	0

Figure 8: *Switching direction of selection control signals for each case of control signal transition in the 3-to-1 multiplexer : (a) order1, (b) order2, and (c) proposed ordering : Dotted boxes show the worst-case transitions.*

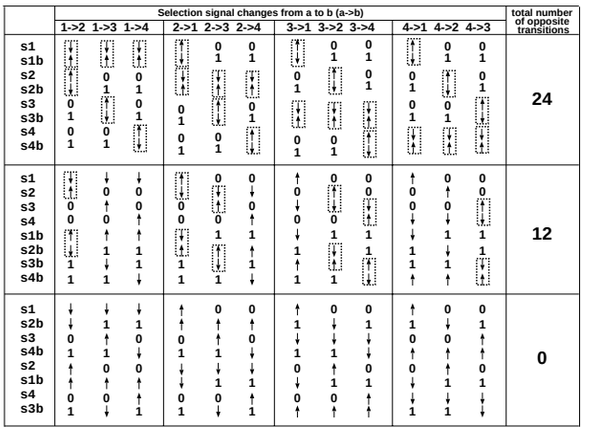

Figure 9: *Switching direction of selection control signals for each case of control signal transition in the 4-to-1 multiplexer : (a) order1, (b) order2, and (c) proposed ordering : Dotted boxes show the worst-case transitions.*

3 Evolutionary Programming-based Datapath Track Assignment Minimizing the Cross-Coupling Effect

The goal of track assignment is to allocate the wire segments into tracks so as to minimize the total number of tracks used. It is known that the following "*Left-Edge algorithm*" gives optimal results[15].

1. Sort the wire segments in the increasing order of their left edges
2. Assign the first segment(the leftmost edge) to the first track
3. Find the first wire whose left edge is to the right of the selected wire and assign it to the current track
4. If no more wire can be assigned to the current track, start a new track and begin again from step 2.
5. Repeat until all wires are assigned to tracks.

In this section, we focus on minimizing the cross-coupling effect in the track assignment. To mitigate the coupling capacitance between long interconnects, we should minimize the common runlength between adjacent wires.

As the cross-coupling in the wires depends on the relative positions of the tracks rather than the absolute locations of the tracks, the track permutation or ordering problem is shown to be NP-complete[8]. To solve the NP-complete problems, the probabilistic hill climbing techniques have been used to escape from the local optimum state. We proposed an "*evolutionary programming-based track assignment(EPTA)*" algorithm considering both the interconnect length and the switching activity. Evolutionary programming(EP) is another branch of the stochastic optimization such as simulated annealing(SA) and genetic algorithm(GA). It was shown that the solution quality of EP is quite good and the converging speed is very fast compared to SA and GA[16, 17]. More than anything else, EP doesn't requires an annoying genetic operator design, which makes the implementation of EP very easy.

As the coupling capacitance is proportional to the length L of wire segments, and inversely proportional to the spacing between wires[7], the cost can be simply calculated by the following formula:

$$Cost = \alpha_1 * NumberOfTracks + \alpha_2 * \max_{ith\ net} Coupling_i \quad (1)$$

$$Coupling_i = \beta_0 * L_{0i} + \beta_1 * L_{1i} + \beta_2 * L_{2i} \quad (2)$$

where $\beta_0 = 1.0, \beta_1 = 1.5, \beta_2 = 2.0$. In eq(2), L_{0i}, L_{1i}, and L_{2i} denote the length of segment i having zero, single, and two neighboring net(s), respectively. As the length of interconnect wire is fixed, the resistance is not changing with the track assignment. The drive strength and the timing slack for each net is not considered in the current algorithm implementation.

In the EP approach, existing track assignment is perturbed until it approaches the global optimum. Fig. 10 shows three terminologies, such as "*permutation*", "*swap*", and "*move*". At first, the initial tracks, determined by left-edge algorithm, are permuted in pairs. As further perturbations, two nets which are placed in two different tracks are swapped, and then a specific net can be moved to another track. The validity of the new track assignment should be checked, *i.e.*, any nets should not be overlapped in the same track. In this simple example, the initial cost of 12.8 was reduced to 9.8 with three perturbations. As much as 23% of the interconnect delay, which corresponds to the cross-coupling effect, is reduced by the optimal track assignment.

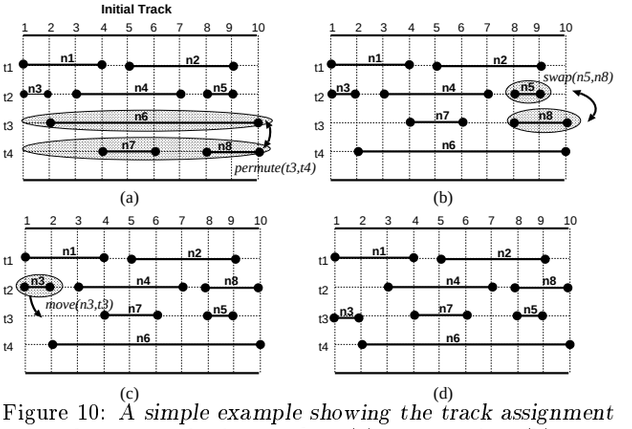

Figure 10: *A simple example showing the track assignment using three perturbations such as (a) permutation, (b) swap, and (c) move, and (d) final result.*

Pseudo code of EPTA algorithm is shown in Fig. 11. An iteration counter is used for the convergence and is determined according to the result of perturbation. If the perturbation leads to a better solution than the current best solution, the iteration counter is decreased, which gives a chance to find a better solution by more computation time. Even though the changes are unprofitable, the perturbation is accepted with finite probability to escape from the local minimum. Consequently, the quality of the final solution obtained improves with iterations.

4 Experimental Result

4.1 Control signal ordering

To demonstrate the effectiveness of the proposed ordering scheme, we experimented with several real microprocessor design examples. For the 64-bit datapath having 15 tracks for each bit slice, the interconnect length for the control wire becomes 1000μm, and the width and space is selected

```
EPTA( )
{
    STEP=5;
    bestCost = LeftEdge( );
    counter = 0;
    while ( counter < MAX_ITERATION ){
        for each Track {
            newCost = perturbation(permute,swap,move);
            check the track overlap;
            if ( Cost is improved ) {
                update bestCost with newCost;
                decrease the counter by STEP;
            }
            else {          /* Cost is not improved */
                if ( random( ) < probability ) {
                    update bestCost with newCost;
                    decrease the counter by STEP;
                }
                else {
                    discard the perturbation;
                    increase the counter by STEP;
                }
            }
        }
    }
```

Figure 11: *Pseudo code of EP(Evolutionary Programming)-based Track Assignment.*

as minimum size in 0.25μm CMOS process. Table 2 shows the typical values of delay and current corresponding to the transition arcs in Fig. 5(a). Since the chip frequency is determined by the longest delay, the maximum delay is shown as the *typical* value, while the average current is shown as typical value in Table 2. Both the delay and current of the "*proposed*" ordering scheme are smaller than those of "*order1*" and "*order2*" scheme.

Table 2: *Delay and switching current for each transition for three ordering schemes in 3-to-1 multiplexer according to 0.25μm CMOS process.*

tran-sition	delay(ps)			switching current(mA)		
	order1	*order2*	*proposed*	*order1*	*order2*	*proposed*
t_{12}	480.95	522.87	401.55	12.128	13.038	10.251
t_{13}	503.01	414.32	413.88	11.866	10.851	10.853
t_{23}	481.54	510.72	402.04	12.640	12.269	10.720
t_{21}	467.99	489.99	382.68	12.171	12.521	10.798
t_{32}	490.65	412.79	412.24	12.700	10.750	10.751
t_{31}	502.61	526.09	412.42	11.534	13.086	10.755
Typ.	503.01	526.09	413.88	73.039	72.515	64.128
Norm.	1.000	1.046	0.823	1.000	0.993	0.878

Table 3 shows the experimental results on several example circuits, which include cascaded 3-to-1 and 4-to-1 multiplexers. The difference between the examples are the multiplexer transistor size and the control buffer size. In average, the multiplexer delay is improved by 14.3% and the average current is reduced by 9.6%.

4.2 Track Assignment

Table 4 summarizes the track assignment result in minimizing the cross-coupling effect. Except for the smallest example, `jhangdp`, which was taken from [7], all examples came from real microprocessor datapath designs. `64bitfpu` is a 64-bit floating point datapath of two-way super-scalar RISC microprocessor.`k486gunit, k386core, k486sunit` come from 32-bit CISC microprocessor designs[18, 19]. In the calculation of cost, α_1 and α_2 are experimentally set to 0.7 and 0.3, respectively. Compared to the existing left-edge algorithm, EPTA reduces the cross-coupling effect by as much as 41%. The convergence behavior of EPTA is shown in Fig. 12 using the example of `k386core` and `k486gunit`.

Table 4: *Comparison between left-edge algorithm and EP-based track assignment minimizing the cross-coupling effect.*

Example	net	track	cost		CPU time
			Left-Edge	EPTA	(sec)
`jhangdp`	12	6	113	63	< 1
`64bitfpu`	18	9	4398	2601	< 1
`k486gunit`	38	18	90103	54200	< 1
`k386core`	46	15	37182	18835	< 1
`k486sunit`	58	18	78042	47414	< 1
sum			209838	123113	

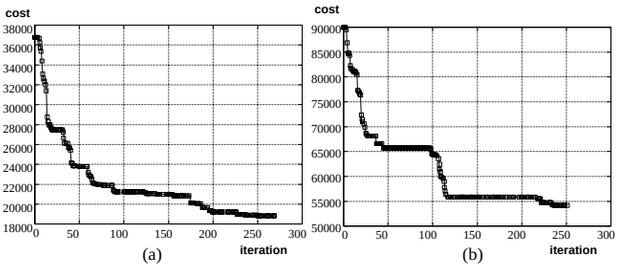

Figure 12: *Convergence behavior of EP-based track assignment on the example (a)* `k386core` *and (b)* `k486gunit`

4.3 Bus Interleaving Considering the Switching Behavior

Above approaches are focused on the geometry, *i.e.*, only the length of overlapped segments is minimized. The cross-coupling effects can be further reduced by carefully interleaving of the busses so that opposite transitions between neighboring wires do not occur simultaneously. One, therefore, needs to pay attention not only to the overlapped length but also to the switching behavior of each signal nets. To reduce the simultaneous transitions in the opposite direction, the following schemes need to be considered.

1) A pair of *heterogeneous phase nets*, *i.e.*, a net switching at *phi1* and the other net at *phi2*, can give shielding effect for each other at each clock phase. Therefore, heterogeneous phase nets should be placed adjacently.

2) *Temporally exclusive nets* also should be placed adjacently. For example, cache *read-bus* and *write-bus* don't switch simultaneously, *i.e.*, they are temporally exclusive. If we interleave these two busses, neighboring nets become quiet for each other, and, therefore, the cross-coupling effect can be eliminated.

3) *Precharged busses* make transition in the uni-direction. All the busses are pulled up in the precharge phase. Therefore, the coupling capacitance does not slow down the transition. During the evaluation phase, all the busses are also guaranteed to make monotonic falling transition. This guarantees that one net will never have to fight against two neighbors that are switching in the opposite direction.

4) *Power/ground bus* also should be considered. Power or ground bus gives good shielding effect on the switching nets. Long running nets should be placed as near the power or ground bus as possible.

For an example in a real 64-bit, RISC microprocessor design, the distance between the source and the destination is about 8000μm as shown in Fig. 13, which is a rather RC-limited time-critical path. In 0.25μm technology, 110ps is a typical value for 1 gate delay(GD). Without any optimization, the propagation delay was initially $23GD$. With setup time of $9GD$, the cycle time is calculated as: $2 \times (23GD +$

Table 3: *The effect of control signal ordering on the delay and average current : improvement(%) = (order1-proposed)/order1*100*

example	delay(ps)			switching current(mA)			improvement(%)	
	order1	*order2*	*proposed*	*order1*	*order2*	*proposed*	delay	power
ex0	486	487	433	147.489	145.354	136.212	10.84	9.94
ex1	526	534	453	155.603	151.214	140.141	13.78	7.65
ex2	502	517	421	75.037	74.341	66.239	16.14	11.72
ex3	614	634	531	93.713	91.795	86.332	13.52	7.88
ex4	652	654	579	161.42	159.99	153.33	11.20	5.00
ex5	527	556	425	72.201	70.974	62.8421	19.39	12.96
ex6	503	526	414	73.039	72.515	64.128	17.69	12.13
ex7	546	570	474	92.013	91.830	83.194	13.29	9.58
ex8	538	557	469	93.673	93.377	84.713	12.79	9.57
average							14.3 %	9.6 %

$9GD$) = $64GD$ = $7.04ns$, corresponding to the clock frequency of 142 MHz. Using the several optimization steps as shown in Table 5, we were able to minimize the delay from $23GD$ to $14GD$, which implies that the cycle time becomes 198MHz. *i.e.*, $2\times(14GD+9GD) = 46GD = 5.06ns$. Among 6 steps in Table 5, steps 3,4, and 6 are based on the bus interleaving schemes proposed in this paper. As a result, total performance improvement was 40%, *i.e.*, from the clock frequency of 140MHz to 200MHz. Above approach is applied to the bus net ordering by hand. But it can be incorporated into the track assignment algorithm as a future work.

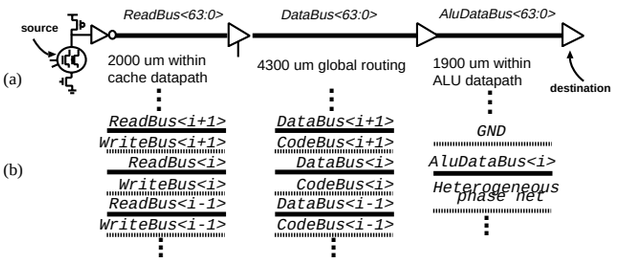

Figure 13: *Example of time-critical 64-bit bus : (a) interconnect segmentation into three parts and (b) bus interleaving.*

Table 5: *Optimization result of time-critical 64-bit bus : By reducing the delay from 23GD to 14GD, the chip performance improves from 140MHz to 200MHz.*

delay	Steps taken to reduce the delay.
$2525ps(23GD)$	1. Initial design
$2306ps(21GD)$	2. Transistor size of domino circuit is optimized.
$1976ps(18GD)$	3. *DataBus* is interleaved with temporally exclusive bus, *CodeBus*.
$1757ps(16GD)$	4. Precharged bus, *ReadBus* is interleaved with *WriteBus*.
$1647ps(15GD)$	5. Intermediate buffers are optimized.
$1537ps(14GD)$	6. *AluDataBus* is placed between ground net and heterogeneous phase net.

5 Conclusion

In this paper, we proposed two interconnect design schemes for minimizing the cross-coupling effect with experimental data based on 0.25μm CMOS technology. To mitigate the cross-coupling effect, the signal switching in the opposite direction should be suppressed both for the control signals as well as the data signals in the custom datapath. Proposed *control signal ordering scheme* was shown to minimize the power consumption by 10% and the delay by 15% for a given set of benchmarks. Furthermore, *evolutionary programming-based track assignment* scheme improves the delay performance by 40%. These schemes were successfully applied to the real microprocessor designs.

Acknowledgement

The constructive comments from the referees were greatly appreciated. I cordially express my gratitude to Ed, Peter, and David who took time off their busy work and provided many helpful discussions joyfully at SandCraft. Also this work may have been impossible without the help of Wooseung and Hunseung in the deep submicron study group in KAIST.

References

[1] Semiconductor Industry Association, *National Technology Roadmap for Semiconductors*, 1994
[2] A.B.Kang *et al.*, "Interconnect Tuning Strategies for High-Performance ICs", *Proc. DATE*, pp.471–478, 1998
[3] D.Li *et al.*, "A Repeater Optimization Methodology for Deep Submicron, High-Performance Processors", *Proc. ICCD*, pp.726–731, 1997
[4] C.D.Kibler, Personal communication on "Interconnect design", SandCraft Inc. 1997
[5] K.Chaudhary, A.Onozawa, and E.S.Kuh, "A spacing algorithm for performance enhancement and cross-talk reduction", *Proc. ICCAD*, pp.697–702, 1993
[6] T. Gao and C.L.Liu "Minimum Crosstalk Channel Routing", *IEEE Trans. CAD-15*, pp.465–474, May. 1996
[7] K.S.Jhang *et.al.*, "COP: A Crosstalk OPtimizer for Gridded Channel Routing", *IEEE Trans. CAD-15*, pp.424–429, Apr. 1996
[8] A.Vittal and M.Marek-Sadowska, "Crosstalk Reduction for VLSI", *IEEE Trans. CAD-16*, no.3, pp.290–298, March 1997
[9] T.Xue *et al.*, , "Post global routing crosstalk synthesis", *IEEE Trans. CAD-16*, no.12, pp.1418–1430, Dec. 1997
[10] A.Onozawa *et al.*, , "Performance driven Spacing Algorithm Using Attractive and Repulsive Constraints for Submicron LSI's", *IEEE Trans. CAD-14*, no.6 pp.707–719, Jun. 1995
[11] H.Zhou and D.F.Wong, "Global Routing with crosstalk constraints", *Proc. 35th DAC*, pp.374–377, June, 1998
[12] H.-P.Tseng, L.Scheffer, and C.Sechen, "Timing and Crosstalk Driven Area Routing", *Proc. 35th DAC*, pp.378–381, June, 1998
[13] S.S.Lai and W.Hwang, "Design and Implementation of Differential Cascode Voltage Switch with Pass-Gate(DCVSPG) Logic for High-Performance Digital Systems", *IEEE JSSC* , Vol.32, No.4, pp.563–573, April, 1997
[14] D.Carlson, *et al.*, "Multimedia Extension for a 550-MHz RISC Microprocessor", *IEEE JSSC* , Vol.32, No.11, pp.1618-1624, Nov., 1997
[15] A.Hashimoto and J.Stevens, "Wire Routing by Optimizing Channel Assignment within Large Apertures", *Proc. 8th DAC*, pp.155–169, June, 1971
[16] C.Sechen,"An improved simulated annealing algorithm for row-based placement", *Proc. ICCAD*, pp.478–481, 1987
[17] Z.Michalewicz, "Genetic Algorithms + Data Structures = Evolution Programs", *Springer-Verlag*, pp.16–17, 1992
[18] C.M. Kyung *et al.*, "HK386: An x86-Compatible 32bit CISC Microprocessor", Proc. *ASP-DAC '97*, pp.661–662, 1997
[19] J.S.Yim *et al.*, "A C-Based RTL Design Verification Methodology for Complex Microprocessor", *Proc. 34th DAC*, pp.83–88, June, 1997