Routing on Regular Segmented 2-D FPGAs

Yu-Liang Wu*

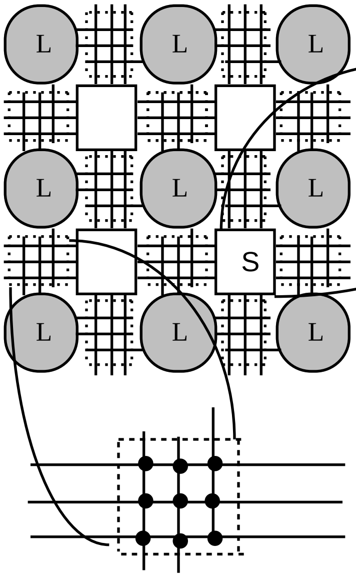
*Cadence Design Systems, Inc.
San Jose, CA. 95134. USA
ylw@cadence.com

Malgorzata Marek-Sadowska+

+Department of Electrical and Computer Engineering
University of California, Santa Barbara, CA 93106.
mms@drum.ece.ucsb.edu

Abstract — *In this paper we analyze the properties of the Xilinx-like regular segmentation schemes for 2-D Field Programmable Gate Arrays (FPGAs). We introduce a new notion of architectural level routing decaying effect caused by wiring segmentation. We discuss its routing properties and propose a relative prime number based segmentation scheme for 2-D FPGA architectures. A new FPGA design concept of applying architectural coupling to achieve better routability is also introduced and experimentally justified*

I. Introduction



A connection (C) box with full connection flexibility

Fig. 1. Unit-segmented FPGA routing architecture (Xc4000 like)

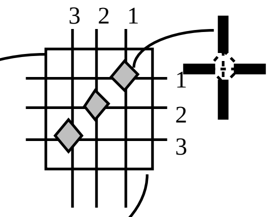


Fig 2a. A disjoint switch box with fs = 1

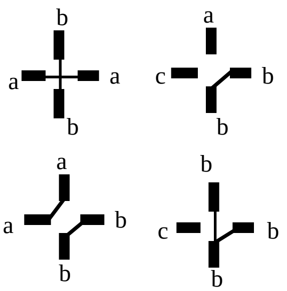


Fig. 2b. The switch (S) box connection examples of each intersection "diamond" shown above

FPGAs (Field Programmable Gate Arrays) are arrays of pre-fabricated logic blocks and wire segments with user-programmable logic and routing resources. Because of their attractive manufacturing cost for low volume production, FPGA usage has grown rapidly for ASIC implementations.

The number of routing tracks of an FPGA chip is limited, and a routing requiring less tracks should have a higher chance of fitting into an FPGA chip. A routing using fewer number of wire segments (and therefore fewer switches) directly reduces the active circuit capacitance and therefore improves the performance and power consumption. These two objectives are related but are not homogeneous.

FPGA routability optimization can be addressed in both algorithmic and architectural aspects. In our earlier research we have shown that simultaneous minimization of tracks and segments requirements is very hard to achieve for an industrial FPGA architecture, and any single cost function based greedy routing algorithm will eventually exhibit a sharp effectiveness decaying effect at the later execution stages on almost every practical circuit example [7]. Regarding the architectural aspects of FPGA routability, most of the previous work assumes randomized (stochastically distributed) segmentation schemes [1,3,9], which inherently lead to difficulties in justifying routability. The segmentation scheme of Xilinx-like FPGA architecture is regular and results obtained for randomized structures do not apply here. In this paper we address the architectural routability optimization aspect for the regular segmentation type FPGAs. In particular we focus on routing architectures whose resources can be partitioned into disjoint domains [2,5,6,8,10] with each domain composed of wire segments of the same length. In the past, routing optimization has been mostly addressed as a resource contention problem, here we show that there is still an overlooked pin topology issue for regular segmented architectures. A pin topology generated without consideration of underlying segmentation may cause poor routing results even under no routing contention.

We will show that an efficiency decaying effect observed in the routing algorithms is also observed on the discussed regular segmented FPGA routing architectures. This is because any such a segmented domain, except the unit-segmented, has certain unroutable connection patterns. Therefore inevitably a routing architecture using uniform long wire segments will suffer a sharp routability decaying effect after its routable net pin position patterns (connection patterns) have been depleted. Based on this observation, we propose a novel approach of segmenting routing domains such that their segment lengths are relatively prime to each other based on the view of enhancing connectable pattern coverage. Here, we also propose a new FPGA chip design concept of applying architectural coupling to achieve better routability. Experimental results justify our new FPGA segmentation scheme.

II. Regular Segmented Fpga Architecture

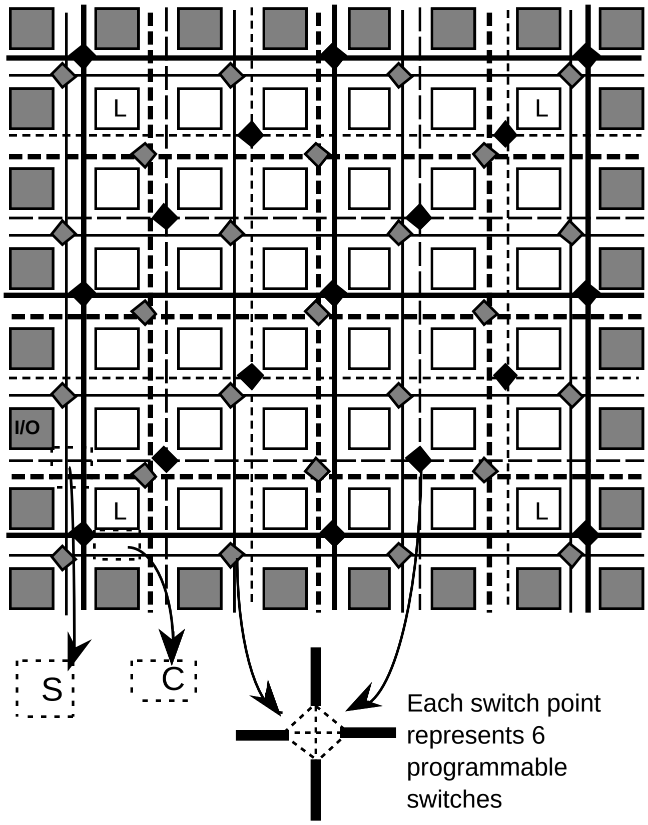


Fig. 3. A regular segmented FPGA with two 2-seg, and three 3-seg routing domains.

To introduce the terminology, we show the simplest unit-segmented routing architecture, XC4000 diagonal S box based in Fig. 1. It is a two-dimensional array of logic cells. Each logic cell is marked L and can be configured to be a Look-Up-Table (LUT), flip-flop,..etc [5,6,7,10]. Wire segments run between the cells in vertical and horizontal channels. Since wires are prefabricated, personalization of routing resources is achieved through a proper programming of switches. Routing switches are grouped inside the connection (C) boxes and switch (S) boxes. The C boxes contain routing switches that can be programmed to connect logic block pins to wire segments. The S boxes contain switches that allow one wire segment to be connected to another.

A vertical (horizontal) *channel* is a sequence of switch boxes, connection boxes and wires between two adjacent columns (rows) of logic cells. The tracks in each vertical (horizontal) channel are numbered from right to left (top to bottom). A number assigned to each track is referred to its *track's id* .

The switching flexibility, f_S, of an S box is the number of wire segments on each of the other three sides of an S box that each segment of one side can connect to.

Here, we assume a logic pin in any C box can connect to any segment running through it, and fs is 1 for every S box. This architecture is disjoint as all connectable routing wires can be partitioned into separate domains, each containing wires of a uniform length. A unit-segment is a segment spanning between two adjacent S boxes and a double-segment is a segment of twice that length. For example, the architecture shown in Fig. 2 is composed of 3 kinds of segment domains: wires of unit length (1-seg), double length (2-seg), and triple length (3-seg). For simplicity, only three 3-seg and two 2-seg routing domains are shown in the figure. In the regular segmented architecture, all connectable wires form a routing domain, and all wires which belong to a common domain have the same length. This is a segmentation style similar to the one adopted in the popular Xc4000 chips [2,10].

III. Computational Complexity of Regular Segmented Routing

The problem of achieving routing with minimum number of tracks, given net pin positions determined by the placer, can be formulated as an optimal augmented graph coloring problem [5,7]. Using too many switches to complete a routing can cause excessive performance degradation. Utilization of longer wire segments is desirable to improve performance.

Here, we formulate the problem of minimum track routing, given a circuit placement as follows:

Problem: [Routing Problem (RP)]

Input: A 2-D FPGA chip with W tracks, pin positions of a set of multi-pin nets determined by a placer.

Decision problem (DRP): Can this routing be completed in W tracks?

Optimization problem (ORP): Find the minimum number of tracks to complete the routing.

In practical routing tasks, completing routing in exactly the minimum number of tracks is not necessarily the most important objective. The routing tracks are prefabricated and as long as the routing can be accommodated, the next goal is to improve the performance of routing. The number and length of segments (and therefore the number of switches) consumed by routing directly affects the active interconnection capacitance, which in turn affects the performance of the whole chip and consumed power. Here, we further consider the objective of completing the routing in the minimum number of segments within the specified track size limit. Below we formulate the problems and analyze their computational complexity.

Problem: [Routing with Segment Usage Bound (RSB)]

Input: A 2-D FPGA chip with W tracks, pin positions of a set of multi-pin nets produced by a placer.

Decision problem (DRSB): Can the routing be completed in W tracks with W*S segments? Here S is the number of segments in a routing track.

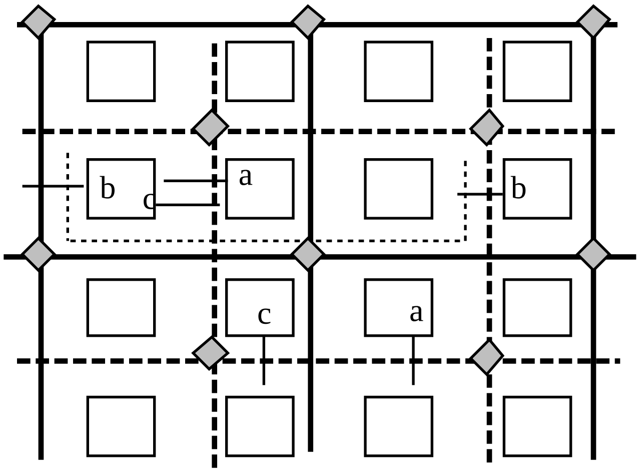
Optimization problem (ORSB): Find the minimum number of routing segments required to complete the routing in specified number of tracks.

The computational complexity of the above problems is given by the following theorems. All these routing problems can be shown to be intractable [5].

Theorem 1. Both [DRP] and [ORP] are NP-hard for regular-segmented architecture.

Theorem 2. Both [DRSB] and [ORSB] are NP-hard for unit length segmented architecture.

IV. Unroutable Patterns and Routability Decaying Effect



a : can be perfectly routed
c : non perfectly routable (with one waste wire un
b : not routable in these domains

Fig. 4. A regular 2-segmented structure

A routing architecture with shorter wire segments (smaller segmentation resolution) has an advantage of finishing routing with fewer tracks due to its higher adaptability to a larger space of connection patterns. However, the number of segments (and also number of switches) used in fine segmented architectures tends to be larger what causes a performance degradation, especially in current FPGA technologies in which the delay caused by switches has a dominating effect. Therefore, using the less flexible longer segments is desirable in current high performance FPGA applications. We will show that any regular segmented domain, except the unit-segmented, has a set of unroutable connection patterns. The existence of an unroutable set will make the routing fail regardless of the number of routing domains.

In Fig. 4, we show examples illustrating the concept of unroutable patterns for a given segmentation. In this figure, there are 3 nets to be routed. Nets *a* and *c* are routable in the dotted segment domain, and net *a* is said to be *perfectly routable* since there exists a pattern connecting both of its 2 pins which does not waste any wire portion (the routing path length is the same as the Manhattan distance of the 2 pins). The net *c* is routable with half wire wasted. However, the net *b* is not routable at all in these domains. This is because any connection pattern for net *b* would require at least one horizontal run with odd length. Such a connection pattern can not be constructed from all even length wires. By the same argument, the net *b* is not routable in domains of 4-seg, 6-seg, or 8-seg, ... etc., since there is no common domain for these 2 pins.

Here, we have a basic property of the regular segmented routing:

Lemma 1. In an m-segmented (m_seg) domain, a 2-pin net is unroutable iff its 2 pins are both located at either horizontal or vertical channels with channel distance ≠ K·m wire units, where K ≥ 0.

In Fig. 5, we show the example topologies of routable and unroutable routing patterns.

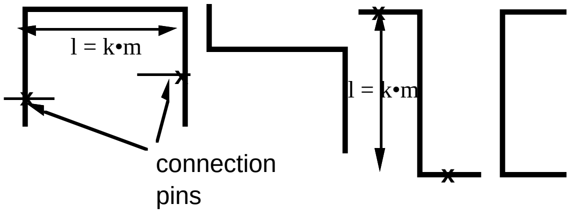


Fig. 5.a. Net topology patterns that require their pin horizontal (vertical) distance be a multiple of m to be routable on an m-seg domain

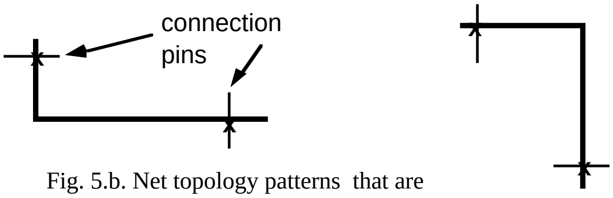


Fig. 5.b. Net topology patterns that are always routable.

Clearly, any non unit-segmented routing structure has a non empty unroutable pattern set. We have the following:

Theorem 3. A regular m-seg structure has its unroutable pattern set dominating any k·m-segmented structure, where $k \geq 1$. Namely, if we denote P_m as the unroutable pattern set of an m-seg domain, then $P_m \subset P_{k \cdot m}$. I.e. if net *i* is not routable in an m-seg domain, then net *i* is also not routable on all other k·m-seg domains.

The theorem implies that these k·m-seg domains do not complement the unroutable set of m-seg domains well.

V. An Effective Segmentation Scheme

In our earlier works [7], it has been shown that by applying the concept of coupling we can substantially improve the results of routing algorithms. In this section, we will show that the concept of coupling can also be applied in hardware design to either improve the chip performance or routability while using fewer routing resources.

A. Relatively Prime Segmentation Coupling

In the regular segmented architecture, similarly to the decaying effect we have observed on routing algorithms, an architecture with too many routing domains of certain segmentations is not likely to be effective. After the supply of routable connections is depleted, another kind of routability decaying effect occurs. The effect is caused by a poor match between the nets pin distribution and connectable pattern sets induced by the segmentation domains.

Table I. Router performance comparison (Xc4000, fs=1, only 1-seg domains)

W	CGE[1]	GGR[7]
BUSC	10	9
DMA	11	9
DFSM	11	10
BNRE	12	11
Z03	14	12
Total	**58**	**51**

W: number of tracks used in routing

Therefore, a good mix between routing domains of different lengths can be viewed as a kind of coupling at architectural level. In our experiment, we compare routing in three types of regular segmentations, a unit-segmented (1-seg), a double-segmented (2-seg), and a triple-segmented (3-seg). Since the average 2-pin net length of our tested circuits is only between 4 to 6 unit segments, we did not experiment with longer wire segments (and the effectiveness of applying 3-seg domains is not significant). In our experiments, we used an efficient greedy algorithm designed explicitly for this kind of disjoint routing domain structures [7]. Table I gives a performance comparison of this router and the well known CGE router [1] on chip with only 1-seg domains. More data can be found in [7].

As seen in the Fig. 6, a typical routing results (circuit: alu4) shown on three one-type segmentation architectures exhibit sharp degradation of routing resource usage. The degradation shown for 1-seg domains is in fact caused by the decaying effect of the routing algorithm itself, since there is no unroutable pattern set for 1-seg domains. The other 2 segmentations show an early decay and leave a large number of unrouted nets. As shown in Fig. 7, after the 10th 2-seg domain, the 2-seg utilization rate drops sharply and none among the remaining nets is routable after the 14 2-seg domains are used. Results of Table 2 show that none of those uniform multi-segmented structures can complete the routing on any circuit, and those unroutable nets are routed by unit-segmented domains resulting in large number of used domains, W.

We can also consider this sharp usage degeneration as an architectural level routability decaying effect, and improve the routability by supplying domains of different segmentation. (Note that since this is an architectural level decaying effect, it can not be cured by simply applying any routing algorithm.) Therefore we have also experimented with other wiring structures. The first one was composed of 7 double-segment domains and the second had 7 triple-segment domains. In both cases, the remaining domains were all of unit length segments. The third structure was a mix of 3 3-seg domains, 4 2-seg domains, and the rest were 1-seg domains. A greedy segmented router similar to [7] was applied on all domains.

The results of this experiment are all shown in Table II. There, we also show the number of switches used by routing. A clear tradeoff can be observed between the number of switches and the number of tracks in multi-segmented architectures. For example, in comparison to 1-seg only domains, using the coupling of 7 double-segmented and unit-segmented routing domains, we can complete the routing with 15% less switches (and segments) although using 22% more tracks. It is surprising to see that the 3-seg/1-seg coupling is not as good as the 2-seg/1-seg coupling, since it is harder to pick 3-seg routable nets, and more nets are left to be routed in 1-seg domains. However, an appropriate 3-seg/2-seg/1-seg coupling can do better than a purely 2-seg/1-seg coupling, especially on larger circuits.

From the experimental results, we can see that the *pin topology factor*, which has been overlooked in previous placement formulations, is playing a significant role in routability of segmented routing architectures.

We note that in a segmentation scheme, which has its different segmented domains designed using relatively prime lengths, some interesting routing properties can be observed. For example, a net with its pin topology perfectly routable in one domain is not likely to be also perfectly routable in another domain with different segmentation (except very long nets). On the other hand, as the chance of mutual domination of unroutable net patterns is minimized (i.e. the coverage of routable patterns are enlarged) a long net should have a higher probability of being accommodated in an m-segmented domain (where m > 1).

B. Switching Flexibility Coupling

The results shown in unit-segmented routing [7], and Fig. 6 and 7, suggest that the early filled domains, even if their switch flexibility fs is just 1 (domain capacity is also 1), achieve high (60~70%) segment usage. Note that the routing segments on the 4 bounding sides are not usable for routing except for connecting very few logic pins at the boundary C boxes. Therefore, this usage is not likely to be improved even with much higher supplies of switches. The reason for achieving high segment usage at early stages is due to the large selection space of unrouted nets. Therefore, at an early stage, a larger supply of switches is most likely just a waste of hardware resources. However, after passing some threshold the low switch flexibility of routing domains (with fs =1) starts to dominate the routing process. At this stage, having routing domains with larger flexibilities would certainly improve the routability.

A direct implication of our experiments is that the switching flexibility in Xc3000 chips (fs = 2 everywhere) is most likely a wasteful resource arrangement. An architecture composed of domains with different capacities should be able to achieve similar routability while using fewer total switches. For example, the earlier packed domains can be those of Xc4000 capacity, and the latter can be the doubled flexibility Xc3000 domains.

C. Architecture Dependent Design Automation

The analysis results of the regular segmented routing also have a direct implication on the placement step. In order to ensure that the critical net pins are placed on long-segment routable domains, the segmentation information must be given to the placer, and the placer should make explicit use of this information to avoid creating unroutable net topologies. For example, the placer may want to avoid placing two critical net pins both on either vertical or horizontal channels unless the distance of these two channels is a multiple of the segmentation length. This objective must also be taken care of during the multiple net decomposition step. We suspect that the higher level design automation steps may also find similar architecture dependent (or technology dependent) optimization issues.

VI. CONCLUSIONS

In this paper we first report an observation of the routability decaying effect on regular one-type non-unit segmented routing domains. We show that this phenomenon is caused by unroutable patterns for each segmentation and improper matching between long net pin topology and wire segmentation. We found that this overlooked factor is dominating on regular segmented routing architectures. To avoid mutual domination of the unroutable patterns and therefore better utilization of long segments, we propose a relatively prime length based segmentation scheme.

Specifically, we find that by applying the approach of architectural coupling, routability can be improved in hardware aspect too. The results also suggest the need of binding together the routing strategies and the underlying architecture properties for better performance. We hope that this notion may serve as a hardware engineering improvement scheme.

VII. ACKNOWLEDGMENTS

The authors would like to thank Professor J. Rose and Professor S. Brown from the University of Toronto for supplying us with the placed circuits used in our experiments. This work was partially supported by the NSF grant MIP91-17328 and MIP94-19119 and in part by AT&T and Xilinx through the California MICRO program.

REFERENCES

[1] G. Lemieux and S. Brown, "A Detailed Routing Algorithm for Allocating Wire Segments in FPGAs", *4th ACM/SIGDA Physical Design Workshop*, pp. 215-226, 1993.

[2] H. Hsieh, et. al. "Third-Generation Architecture Boosts Speed and Density of Field Programmable Gate Arrays", *CICC*, pp. 31.2.1-31.2.7. 1990.

[3] Y. Sun, T.C. Wang, C.K.Wong and C.L.Liu, "Routing for Symmetric FPGAs and FPICs", *ICCAD*, pp. 486-490, 1993

[4] S. Trimberger, "A Reprogrammable Gate Array and Applications", *Proc. of The IEEE*, pp. 1030-1041, 1993.

[5] Y.L. Wu "On 2-D FPGA Routing: Theoretical Analysis and Novel Effective Solutions", Ph. D. thesis, ECE department, University of California Santa Barbara, 1994.

[6] Y.L.Wu and D. Chang, "On NP-Completeness of 2-D FPGA Routing Architectures and a Novel Solution", *ICCAD* pp. 362-366, 1994.

[7] Y.L. Wu and M. Marek-Sadowska, "Orthogonal Greedy Coupling - A New Optimization Approach to 2-D FPGA Routing", *DAC* pp. 568-573. 1995.

[8]. Y.L. Wu, S. Tsukiyama, and M. Marek-Sadowska, "On Computational Complexity of a Detailed Routing Problem in Two-Dimensional FPGAs", *4th Great Lakes Symposium on VLSI*, March 1994.

[9] K. Zhu, D. Wong and Y Chang, "Switch Module Design with Application to 2-D Segmentation Design", *ICCAD*, pp.480-485, 1993.

[10] "The Programmable Logic Data Book", *Xilinx*, 1994.

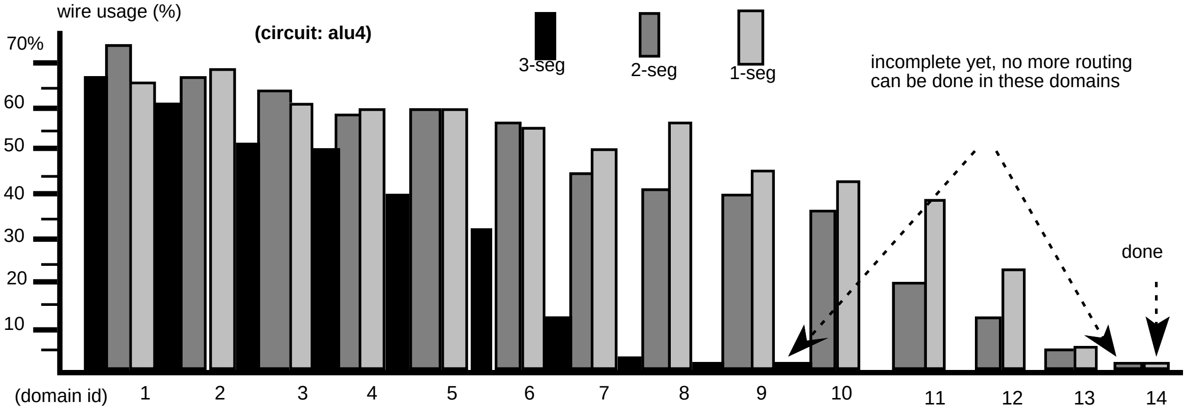


Fig. 6. Wire usage(%) on purely 3-seg, 2-seg, or 1-seg architectures. All exhibit sharp decays. On 3-seg and 2-seg structures, a large number of nets are not routable, routing is incompleted.

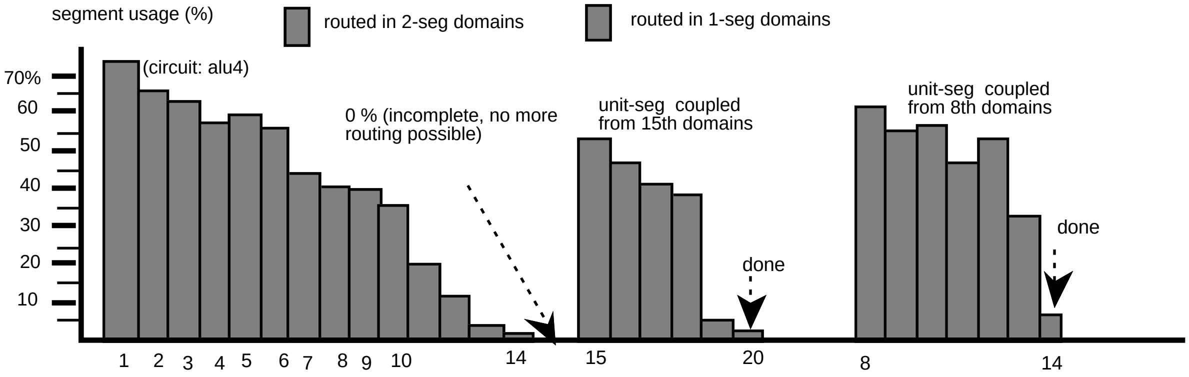


Fig. 7. Segment usage(%) on purely double-segmented architecture and architecture coupled with unit-segmented domains

Table II. Routing results for single segmented or mixed coupling architectures

Circuits	All 1-seg dom		*Mostly 3-seg dom		*Mostly 2-seg dom		3-seg: 7 doms + 1-seg: ~		2-seg: 7 doms + 1-seg: ~		3-seg: 3 doms + 2-seg : 4 doms, +1-seg: ~	
	W	|sw's|	W	|sw's|	W	|sw's|	W	|sw's|	W	|sw's|	W	|sw's|
alu4	12	4819	20	4195	20	3747	17	4254	14	4029	15	4019
apex7	10	1469	18	1294	18	1118	16	1269	12	1218	13	1194
example2	12	2541	19	2233	20	2068	17	2253	15	2224	16	2151
too_large	11	2796	16	2536	18	2198	16	2524	14	2312	14	2299
k2	16	8536	21	7224	24	6428	19	1256	18	7348	17	7186
vda	12	4063	18	3645	20	3094	17	3559	15	3407	15	3403
alu2	11	2445	14	2212	15	1960	16	2235	12	2059	14	2104
term1	9	904	14	871	14	738	14	858	12	752	12	776
9symml	9	1074	15	993	14	873	14	1004	11	900	11	921
apex6	16	6332	24	5608	27	4846	21	5631	18	5450	19	5387
cht	11	976	19	849	20	777	16	898	13	847	15	825
C880	11	2314	16	2070	18	1813	17	2070	14	1959	14	1929
decod	6	289	9	246	11	257	11	243	10	256	10	232
Total	146	38558							178 (+22%)	32761 (-15%)	185 (+26.7%)	32603 (-15.4%)

*Mostly 2-seg dom: All nets are first tried to be routed on 2-seg domains until not routable any more. The left unroutable nets are then routed on 1-seg domains.