Evaluation of Dual V_{DD} Fabrics for Low Power FPGAs

Rajarshi Mukherjee and Seda Ogrenci Memik
Electrical and Computer Engineering Department, Northwestern University
{rajarshi, seda} @ece.northwestern.edu

Abstract - Power efficiency is becoming an increasingly important design aspect for FPGAs. Recently it has been shown that well-known power minimization techniques in the ASICs such as creating supply voltage (V_{dd}) scalable islands of different granularity can be applied to FPGAs. However, the discrete routing architecture of FPGAs amplifies any constraint imposed on the placement stage. In this work, we evaluate the overheads of voltage scaling schemes in relation to FPGA architectures and design flows in terms of critical path delay, channel-width and area/delay product. We present a detailed evaluation of the impact of alternative realizations of voltage scaling schemes onto the physical design flow of FPGAs and show that as high as 47% dynamic power gain is possible with 17% area/delay product penalty and 30% power gain is possible with as low as 6% area/delay product penalty for different voltage island configurations.

1. Introduction

Dynamic power is proportional to V_{dd}^2 and leakage has an exponential dependency on V_{dd}. On the other hand, delay increases approximately linearly as the supply voltage is decreased. Hence, it is possible to perform a tradeoff between power consumption and performance by changing the supply voltage. Logic blocks that lie on the critical path (longest combinational path) are called *critical nodes*. Any increase of the delay of a critical node would result in an increase of the critical path delay and of the overall circuit. Remaining nodes are referred to as *non-critical* nodes and have positive time slack. Hence, it is possible to identify the inherent timing freedom possessed by non-critical nodes in a design and operate them at a lower V_{dd} still maintaining the performance. On the other hand, slowdown of logic due to scaling of the supply voltage can be avoided by simultaneously reducing the threshold voltage V_t. However, this cannot be done beyond a certain limit since the increase in leakage power due to reduction of threshold voltage would start to dominate.

Voltage scaling is a well-known tool for improving energy efficiency of microprocessors, ASICs and real time embedded systems [1-5]. In ASICs it has been shown that dual-V_{dd} achieves more power reduction than by V_{dd} scaling for a given performance constraint [6, 7]. Dual threshold (Dual-V_t) techniques have been applied to ASICs to reduce leakage power [8].

Despite exponential improvements in logic density and performance, energy efficiency of the FPGA technology did not keep up. An experimental study comparing the energy consumption of an 8-bit adder implemented in a Xilinx XC4003A FPGA with that of a customized CMOS implementation showed a 100x difference in favor of ASIC implementation (4.2mW/MHz at 5V for FPGA vs. 5.5uW/Mhz at 3.3V for ASIC counterpart) [9]. Most of the FPGA synthesis techniques address power optimization by improving metrics such as switching activity, total logic resources used, interconnect used, etc. Technology mapping targeting power minimization in FPGAs have been proposed [10-12].

Recently there has been lot of interest in dual-V_{dd} / dual-V_t FPGA architectures. Li et al. first reported power reduction using dual supply /dual threshold voltage for pre-defined FPGA fabrics [13]. In addition Chen et al. proposed a low-power technology mapping technique for dual V_{dd} FPGAs [14]. Subsequently power reduction using programmable dual V_{dd} fabrics has been proposed [15]. A power driven partitioning algorithm was introduced to partition a netlist onto predefined voltage islands on FPGA [16]. Previous works proposed effective techniques to realize dual V_{dd} fabrics for significant power gain. However, they did not offer a detailed evaluation of the impact of their techniques onto final quality of the design in terms of area and delay overhead. Li et al. [13] performed an evaluation aiming the same operating frequency before and after V_{dd} scaling. In practice a designer may want to do this often, i.e. apply V_{dd} scaling only to the extent where it does not affect the original frequency of operation. However, there is still a need for more insight into the nature of the trade-off beyond that limit.

Starting off with the motivation described above, the crucial task is to evaluate the overheads and impacts of voltage scaling schemes on the FPGA architectures and design flows. In this work, we focus on the issues related to realization of dual-V_{dd} FPGA fabrics. The discrete routing architecture of FPGAs amplifies any constraint placed on the placement stage. Therefore, the restrictions imposed on placement by the distribution of supply voltage across the chip have a significant impact on the quality of the design in terms of critical path delay, channel width, and total area. We present an extensive evaluation of the impact of alternative dual supply voltage distribution topologies on the physical design stage and report the associated power savings. For the rest of the paper we adopt a model, where the threshold voltage is not changed and the trade-off is performed between delay and supply voltage level.

The rest of the paper is organized as follows. In Section 2 a brief overview of the power-driven partitioning algorithm that we used to create high/low V_{dd} voltage islands. In Section 3 we discuss the implementation issues of different voltage island topologies. Our experimental setup and results are presented in Section 4 and we conclude with a summary in Section 5.

2. Overview of Power Driven Design Partitioning

Given a design, power driven partitioner creates clusters of critical and non-critical nodes. The aim is to increase delay of the longest path in some non-critical node clusters by voltage scaling with consequent power reduction and without affecting the overall delay of the circuit. The slack of longest path within that cluster will dictate the voltage-scaling factor. The algorithm takes as input a netlist of CLBs obtained after technology mapping and packing stages and a feasible scale factor *Sf* – the minimum amount by which voltage can be scaled. An overview of the algorithm is given in Figure 1. CreateCluster procedure is invoked for creating clusters that can run at reduced voltage. If a node has zero slack it is added to the non-scaled partition. After choosing a non-zero slack node *v*,

which is not already added to any partition, it is checked whether the scale factor of the node is greater than input *Sf*. If this condition is satisfied, then a new cluster is created with this seed node and the algorithm tries to grow this cluster by adding nodes along a path.

```
Power Driven Partitioner (Netlist G, Sf,
                          Resource Constraint)
Output: Voltage Scalable Clusters
Start
    Topological Sort(G)
    CP = Length of Critical Path
    Compute Required and Arrival Times
    ASAP(G)
    ALAP(G, CP)
    Compute Slack of Nodes S = Required - Arrival
    CreateCluster(G)
    Merge Clusters to Create larger Clusters based on
    Resource Constraint
End
```

Figure 1. Overview of power-driven partitioning algorithm.

The CreateCluster procedure is repeated until all nodes in the input circuit are assigned to some cluster. Note this estimation of the slack of each logic block is at a high level and an accurate estimation of interconnect delay cannot be made before physical synthesis. In the case of the single FPGA system, due to the overhead of creating voltage islands on a chip, we create voltage islands supplied by only two levels V_{dd}^{high} or V_{dd}^{low}. If the number of clusters exceeds this threshold, multiple clusters are merged with critical nodes being in the V_{dd}^{high} island and non-critical nodes in the V_{dd}^{low} island. After assignment of clusters to FPGAs, the total power consumption becomes P_{scale}. If the total power consumption without any voltage scaling was P, the ratio is given by

$$\frac{P_{scale}}{P} = \frac{C_H(V_H)^2 + C_L(V_L)^2}{C(V_H)^2}$$

where C_L and C_H refer to the total capacitance of low V_{dd} CLBs and high V_{dd} CLBs respectively. A similar formulation has been previously proposed by Hamada et al. in [7].

3. Implementation Issues for Alternative Voltage Island Topologies

Creating voltage islands on a single FPGA will incur certain hardware costs. We perform a qualitative comparison of various topologies for two main cost criteria: level converters and power grid complexity. Voltage islands can be supplied by either V_{dd}^{high} or V_{dd}^{low}. The location of the voltage islands is an architectural parameter. Figures 2(a), 2(b), and 2(c) show 3 possible fabrics presented in [16]. In [13], row based and interleaved dual-V_{dd} fabrics have been proposed shown in Figures 2(d) and 2(e). For the fabric in Figure 2(e) the level of granularity is a single CLB.

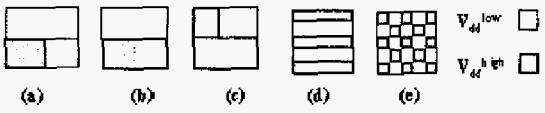

Figure 2. Proposed voltage island configurations.

Assuming interleaved layout pattern using configurable/programmable dual V_{dd} CLBs has also been proposed in previous works [15]. A FPGA having dimensions D_X, D_Y has total $(D_X \times D_Y)$ logic blocks. Our topologies are characterized by f – the fraction of *critical nodes* to total logic blocks. Fabric 2(a) is used if is $f \leq 25\%$, 2(b) is used if $25\% \leq f \leq 50\%$ and 2(c) is used if $50\% \leq f \leq 75\%$. Obviously for $75\% \leq f \leq 100$, all the quadrants must be supplied with V_{dd}^{high} and there is no power improvement possible by V_{dd} scaling. For row-based fabric the ratio of V_{dd}^{high} row: V_{dd}^{low} row can be 1:3, 1:2 and 1:1 for $f \leq 25\%$, $25\% \leq f \leq 33\%$, and $33\% \leq f \leq 50\%$ respectively. For the interleaved topology the ratio of V_{dd}^{high} CLBs: V_{dd}^{low} CLBs is set as 1:1.

3.1 Placement of Level Converters

To avoid excessive static power consumption level converters (LC) need to be inserted when CLBs in the V_{dd}^{low} region drive CLBs in the V_{dd}^{high} region, which converts V_{dd}^{low} swing to V_{dd}^{high} swing. Level converters are not needed when CLBs in the V_{dd}^{high} region drive CLBs in the V_{dd}^{low} region. Puri et al have proposed an efficient level converter design, which uses a single supply (V_{dd}^{high}) voltage [17]. Here, we are considering the granularity level to be CLBs. For CLBs, Li et al. have proposed using LCs at the output pins of CLBs in the V_{dd}^{low} regions used to drive CLBs in the V_{dd}^{high} region [13]. In the experimental section we will show that critical: non-critical CLB ratio for a set of 10 MCNC benchmarks is 1:6.25 (i.e. 16% of the total CLBs is critical). The circuits can be placed and routed for the FPGA fabrics shown in Figure 2(a) or 2(d) (with V_{dd}^{high} CLBs being less than 25%). In that case, it is attractive to place LCs at the inputs of the CLBs in the V_{dd}^{high} region. This enables non-critical CLBs to be supplied by V_{dd}^{low} only. Otherwise if LCs are placed in CLBs in V_{dd}^{low} region, CLBs have to be also supplied by V_{dd}^{high} for their LCs potentially increasing the complexity of the power grid.

3.2 Power Grid

The logic power grid distribution for Xilinx XC4000 FPGA shown in Xilinx XC4000E product datasheet is presented in Figure 3(a). For the fabric in Figure 2(b) with LCs in the V_{dd} high region, the dual Vdd routing is simpler – with one half being supplied by V_{dd}^{high} and other by V_{dd}^{low} as shown in Figure 3(b). For fabric in Figure 2(a) and 2(c), for half of the FPGA, the power lines are V_{dd}^{high} (V_{dd}^{low}) only and in the other half V_{dd}^{high} and V_{dd}^{low} power lines run together. For row-based fabric in Figure 2(d), the V_{dd}^{high} and V_{dd}^{low} power lines are interleaved across the full chip (Figure 3c). For the interleaved fabric in Figure 2(e) with predefined or programmable high and low Vdd CLBs the power grid routing will be most complex where V_{dd}^{high} and V_{dd}^{low} power lines have to run together supplying each row with both V_{dd}^{high} and V_{dd}^{low} as shown in Figure 3(d).

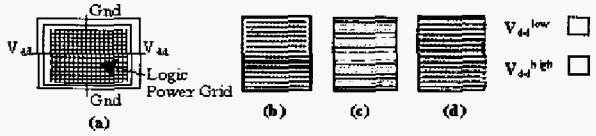

Figure 3. Logic power grid topologies.

4. Experiments

We present our experimental results in this section. The experiments are formed on a set of MCNC benchmarks [18]. We used T-VPack [19] to pack LUTs with clusters of size 10 and 22 inputs per cluster. For other popular architecture configurations, such as a cluster size of 4 with 10 inputs per cluster, we observed similar trends throughout our experiments. The power-driven partitioner reads in the output of T-VPack and produces 2 voltage scalable clusters. The clusters are then placed and routed onto voltage islands on a single FPGA using VPR. We have implemented additional constraints on the placer to ensure the proper locations of CLBs into

the respective voltage level regions. Finally, the power, area/delay product, channel width and delay are compared against unconstrained placement. We assumed that voltage scaling is done within a range between 0.8V and 1.3V in increments of 0.06 Volts. The minimum feasible scale factor *Sf* is then 0.62 and V_{dd}^{high} is set to 1.3V and Vdd^{low} to 0.8 V. The techniques are independent of the actual values of these parameters; hence, they can operate under different assumptions.

4.1 Dual V_{dd} Voltage Islands on a Single FPGA.

We now present the results for voltage islands supplied by different V_{dd} levels on a single FPGA. The placement is based on the Simulated Annealing (SA) implemented in VPR. The linear congestion cost function used in VPR is enhanced to impose constrained placement. This formulation is also used in previous works [13, 16] for placement restrictions where matched(j) returns 1 if the j^{th} logic block is placed in its matching voltage quadrant/row/CLB and 0 if not. Δmatched is the difference between matched(j) in the previous placement and matched(j) in the present placement and penalizes a move that brings a block from matched to unmatched quadrant/row/CLB; α, γ are appropriate constants.:

$$\Delta C = \Delta C_{linear\_conj} + \alpha\Delta matched(j) + \gamma(1 - matched(j))$$

Table 1 shows the total number of logic blocks, the CLB array dimension $D_X = D_Y$, % *f* (fraction of critical nodes to total CLBs), primary inputs (PI) and outputs (PO) for each benchmark. The Channel Width (CW), Critical Path (CP) delay and % power improvement are shown for unconstrained placement assuming 100% voltage programmable CLBs. The average % power improvement is 51.33%.

We have different FPGA fabrics (as discussed in Section 3) with predefined V_{dd}^{high} regions. Region-Constrained Placement (RCP) places the cluster of critical nodes in a V_{dd}^{high} region determined by *f*. However, the *non-critical* blocks have controlled freedom to move to V_{dd}^{high} quadrant(s) if it improves the congestion cost. Therefore, the final achieved power improvement after constrained placement is less than the maximum achievable improvement. For our benchmarks 16% of the CLBs are critical on average. Table 2 (next page) shows the %CW penalty, %CP Penalty and %Power Improvement for constraining critical CLBs in different types of FPGA fabrics considering Table 1 as the base case. RCP for smaller regions results in non- negligible channel width increase. The VPR tool reports the minimum channel width at which a design could successfully be placed and routed. Naturally, constraints on the placement place a bigger burden on physical design leading to an increase in required channel width for routability. For RCP, we start the simulated annealing algorithm with an initial placement where critical CLBs are in V_{dd}^{high} region(s). Negative values for the penalty (↔gain) indicate two possible cases: 1) Penalty in CW with gain in CP indicates that channel width has increased due to constrained placement, now making the routing relatively easier; 2) Gain in CW with gain in CP indicates simulated annealing has found a better solution than the unconstrained case. Obviously, if simulated annealing is made to run an unbounded number of iterations, then unconstrained placement will give better results. For the benchmarks alu4 and misex3, f > 25%. Hence 25% constrained placement (row/quadrant based) results are not available for them. It can be seen from Table 2 that 50% Row Based, Interleaved, and 50% Quadrant based have least CW penalty. For these, less *non-critical* CLBs can now be in the V_{dd}^{low} region with power gain is in the range of 30% while for 25% Row Based/Quadrant based the power gain is around 40% with higher CW penalty.

Table 1. Placement results for V_{dd} Programmable CLBs

							Unconstrained		
Circuits	CLBs	Critical Nodes	Dx=Dy	% *f*	PI	PO	CW	CP	%Power Imp
Alu4	154	48	13	28	14	8	44	3.89E-08	42.76
apex2	190	27	14	14	39	3	59	3.98E-08	53.30
apex4	132	19	12	13	9	19	57	4.12E-08	53.19
des	160	13	32	1	256	245	31	3.97E-08	57.08
ex1010	480	51	22	11	10	10	74	6.06E-08	55.53
ex5p	110	22	11	18	8	63	62	4.06E-08	49.70
misex3	142	46	12	32	14	14	50	3.60E-08	42.00
pdc	462	41	22	8	16	40	97	6.37E-08	56.62
seq	177	27	14	14	41	35	57	3.69E-08	52.65
spla	374	70	20	17	16	46	76	7.35E-08	50.50
Average				**16**					**51.33**

To evaluate the effect of RCP, we use the routing area-delay product calculated as $2 \times D_X \times D_Y \times CW \times Delay(CP)$. Table 3 shows the area/delay product of the unconstrained placement and %change of area/delay products after RCP for different FPGA fabrics with respect to unconstrained placement. Note that the negative values in the table indicate that in some cases the area-delay product is better than the unconstrained placement.

Table 3. Trade-off between area/delay products.

		Row based			Inter-leaved	Quadrant based	
Circuits	Unconst.	V_{dd}^{H} 25%	V_{dd}^{H} 33%	V_{dd}^{H} 50%	V_{dd}^{H} 50%	V_{dd}^{H} 25%	V_{dd}^{H} 50%
alu4	5.78E-04	N/A	-1.66	-1.66	-12.05	N/A	18.21
apex2	9.21E-04	17.23	-3.73	3.51	1.64	19.93	1.05
apex4	6.76E-04	7.64	15.39	13.04	4.53	12.35	5.01
des	2.52E-03	-22.18	-0.99	-18.03	-8.88	-11.26	-14.23
ex1010	4.34E-03	9.59	-10.09	54.59	-1.36	9.98	5.06
ex5p	6.09E-04	-5.60	-5.91	-7.59	0.54	-8.42	-0.60
misex3	5.19E-04	N/A	9.93	9.93	15.96	N/A	16.01
pdc	5.98E-03	36.76	26.86	41.89	10.11	28.26	-5.98
seq	8.24E-04	-1.87	-8.02	1.50	3.21	8.94	8.23
spla	4.47E-03	-8.74	8.50	-19.06	-18.84	-10.78	-7.77
Average		**17.80**	**15.17**	**20.74**	**6.00**	**15.89**	**8.93**

Figure 4 presents the average of the % penalty of the area/delay products for RCP and % power gain for each fabric w.r.t. unconstrained placement. It can be seen 25% row based and 25 % Quadrant based give almost same results since on average 16% V_{dd}^{high} CLBs are constrained within 25% of the FPGA. 50% Quadrant based and interleaved give the best area/delay product results. However 50% row based is worse than 25% Row based. This is because (from Table 3) there is a decrease in CW compared to unconstrained placement and this is associated with much higher critical path delay penalty.

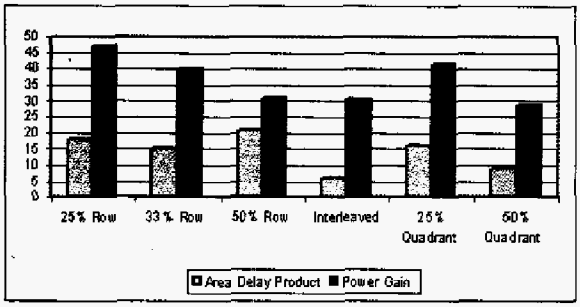

Figure 4. % Change in area-delay product and power improvement for different fabrics.

5. Conclusion

In this paper, we presented comparative evaluation of dual V_{dd} fabrics for low power FPGAs. Dual V_{dd} voltage island technique improves dynamic power of logic blocks and this is what we have evaluated in this work. Using dual V_t technique (e.g. high V_t SRAM cells) is complimentary to using dual V_{dd} and will improve leakage and consequently overall power consumption. We evaluated different configurations of V_{dd}^{high} and V_{dd}^{low} islands for a single FPGA and showed that the creation of voltage islands results in non-negligible penalty in physical design quality in terms of channel width (routing area) and critical path. We also presented associated power gain for constrained placement on different FPGA fabrics. We believe that this study by quantifying tradeoffs between power gain and penalty in other performance metrics (area, delay) is essential to evaluate different realizations of dual V_{dd} techniques.

6. Reference

1. Govil, K., E. Chan, and H. Wasserman, *Comparing Algorithms for Dynamic Speed Setting of a Low Power CPU.* MOBICON. 1995.

2. Iyer, A., and D. Marculescu, *Power Efficiency of Voltage Scaling in Multiple Clock, Multiple Voltage Cores.* International Conference on Computer Aided Design. 2002.

3. Yeh, C., et al, *Gate-Level design Exploiting Dual Supply Voltages for Power-Driven Applications.* Design Automation Conference. 1999.

4. Simunic, T., et al, *Dynamic Voltage Scaling and Power Management for Portable Systems.* Design Automation Conference. 2001.

5. Usami, K., and M. Igarashi, *Low-power Design Methodology and Applications Utilizing Dual Supply Voltages.* Asia and South Pacific Design Automation Conference. 2000.

6. Lackey, D., et al, *Managing power and performance for SoC designs using voltage islands.* International Conference on Computer-Aided Design. 2002. San Jose, CA.

7. Hamada, M., et al, *A Top-Down Low Power Design Technique Using Clustered Voltage Scaling with Variable Supply-Voltage Scheme.* Custom Integrated Circuits Conference. 1998.

8. Kao, J.T., and A.P. Chandrakasan, *Dual-Threshold Voltage Techniques for Low-Power Digital Circuits.* IEEE Journal of Solid State Circuits, 2000. **35(7)**: p. 1009--1018.

9. Kusse, E., and J. Rabaey, *Low-energy embedded FPGA structures.* International Symposium on Low Power Electronics and Design 1998. Monterey, CA., 1998.

10. Lamoureux, J., and S.J.E. Wilton, *On the Interaction Between Power-Aware FPGA CAD Algorithms.* International Conference on Computer Aided Design. 2003.

11. Li, H., S. Katkoori, and W. Mak, *Power Minimization Algorithms for LUT-based FPGA Technology Mapping.* ACM Transactions on Design Automation of Electronic Systems, 2004. **9(1)**: p. 33--51.

12. Wang, Z.H., et al, *Power Minimization in LUT-based FPGA Technology Mapping.* Asia-South Pacific Design Automation Conference. 2001.

13. Li, F., et al, *Low-Power FPGA Using Pre-Defined Dual-Vdd/Dual-Vt Fabrics.* International Symposium on Field-Programmable Gate Arrays. 2004.

14. Chen, D., et al, *Low-Power Technology Mapping for FPGA Architectures with Dual Supply Voltage.* International Symposium on Field-Programmable Gate Arrays. 2004.

15. Li, F., Y. Lin, and L. He, *FPGA Power Reduction Using Configurable Dual-Vdd.* Design Automation Conference. 2004. San Diego, CA.

16. Mukherjee, R., and S. Memik, *Power-Driven Partitioning.* Field Programmable Logic and its Applications. 2004.

17. Puri, R., et al, *Pushing ASIC Performance in Power Envelope.* Design Automation Conference. 2003.

18. Yang, S., *Logic Synthesis and Optimization Benchmarks.* Tech. Report. 1991, Microelectronics Center of North Carolina.

19. Betz, V., J. Rose, and A. Marquardt, *Architecture and CAD for Deep-Submicron FPGAs.*

Table 2. Trade-off between power improvement and penalty in critical path and channel width.

	Row Based V_{dd}^H 25%			Row Based V_{dd}^H 33%			Row based V_{dd}^H 50%			Interleaved V_{dd}^H 50%			Quadrant V_{dd}^H 25%			Quadrant V_{dd}^H 50%		
Circuits	%CW Penalty	%CP Penalty	%Power Imp	%CW Penalty	%CP Penalty	%Power Imp	%CW Penalty	%CP Penalty	%Power Imp	%CW Penalty	%CP Penalty	%Power Imp	%CW Penalty	%CP Penalty	%Power Imp	%CW Penalty	%CP Penalty	%Power Imp
alu4	N/A	N/A	N/A	-2.27	0.62	32.28	-2.27	0.62	32.28	-6.82	-5.61	30.26	N/A	N/A	N/A	0.00	18.21	31.07
apex2	20.34	-2.58	48.40	-3.39	-0.35	44.15	-3.39	7.14	30.74	0.00	1.64	31.07	8.47	10.56	43.82	0.00	1.05	31.72
apex4	14.04	-5.61	46.13	31.58	-12.30	41.89	21.05	-6.62	31.54	7.02	-2.32	31.07	10.53	1.65	43.30	7.02	-1.88	29.65
des	-22.58	0.52	44.66	-25.81	33.45	40.00	-19.35	1.64	29.51	-19.35	12.99	28.74	-12.90	1.88	26.41	-22.58	10.79	13.98
ex1010	4.05	5.32	47.89	1.35	-11.29	42.33	1.35	52.53	31.19	0.00	-1.36	31.07	16.22	-5.37	45.04	9.46	-4.02	30.94
ex5p	-1.61	-4.05	49.70	-1.61	-4.37	43.49	-8.06	0.52	32.76	3.23	-2.60	31.07	-8.06	-0.38	41.23	3.23	-3.71	31.63
misex3	N/A	N/A	N/A	16.00	-5.23	31.07	16.00	-5.23	31.07	-2.00	18.33	30.63	N/A	N/A	N/A	12.00	3.58	30.63
pdc	0.00	36.76	47.61	-6.19	35.22	41.96	-7.22	52.93	31.07	-2.06	12.43	30.93	2.06	25.67	44.92	-1.03	-5.00	30.80
seq	0.00	-1.87	48.09	-3.51	-4.68	43.18	-3.51	5.19	31.59	0.00	3.21	30.89	10.53	-1.43	42.82	7.02	1.13	28.78
spla	2.63	-11.08	46.51	9.21	-0.66	43.03	3.95	-22.14	31.23	6.58	-23.85	31.23	11.84	-20.23	43.86	7.89	-14.52	29.74
Average	**2.11**	**2.18**	**47.37**	**1.54**	**3.04**	**40.33**	**-0.15**	**8.66**	**31.30**	**-1.34**	**1.28**	**30.69**	**4.83**	**1.54**	**41.42**	**2.30**	**0.56**	**28.89**