LOSP PERFORMANCE EVOLUTION

William Bricken
October 1999

SUMMARY 10-99

The Losp benchmark suite currently consists of over two hundred logic synthesis circuits and circuit fragments from ISCAS'91 (~120 combinational and ~80 sequential circuits, ranging from very small to over 20K gates). These circuits consist of small test cases, many difficult or tricky circuits, and some larger alu and arithmetic examples that are known for their difficulty to optimize. Losp successfully reduces all of them, whether in multilevel or two-level form.

The more challenging benchmarks have published reduction results for several logic synthesis research tools. In comparison to the best performance across all other surveyed techniques, Losp currently outperforms the field on 17 out of 34 circuits, when circuit complexity is measured by the literal count at all gates (that is, the number of input pins across all gates in the design). Losp does slightly better still on other metrics such as gate count and wire count. ***Appendix I*** summarizes these results.

DETAILS 10-99

Four systems with published results for the more difficult ISCAS'91 combinational circuits were used for comparison. 34 circuits were identified for which performance results are available for at least two of the comparison systems.

The four comparison systems were each developed over years by many contributors. HANNIBAL, CLS, and SIS are university projects, while MIRACLE_C was developed at AT&T Bell Labs. At a blind guess, I'd say that each of these systems has an order of magnitude more development effort than Losp.

Comparison Systems

HANNIBAL

Multilevel logic optimization using implication based recursive learning techniques, in
W. Kunz and D. Stoffel (1997) Reasoning in Boolean Networks, Kluwer, p155.

Considered to be the best current research tool for combinational logic optimization.

CLS

The Concurrent Logic Synthesis tool based extensively on combinations of types of BDDs, in
J. Bullmann and U. Kebschull (1996) "Multiple Domain Logic Synthesis", in T. Sasao and M Fujita (Eds) Representations of Discrete Functions, Kluwer, p.227.

One of the best tools using BDDs as the primary circuit representation. CLS switches between several BDD formats to handle the diversity of circuit forms, some of which grow exponentially in specific BDD notations.

MIRACLE_C

The logic redundancy removal system based on fault identification using iterative addition and removal of circuit elements, in

K-T Cheng and L. Entrena (1993) "Multi-Level Logic Optimization By Redundancy Addition and Removal", in IEEE 1066-1409/93.

This technique represents work from the Automatic Test Pattern Generation community, and reflects non-algebraic methodologies that rely on testing specific instances for circuit redundancy.

SIS

A fourth standard reference tool is SIS, the sequential multilevel synthesis tools provided by UC Berkeley.

All of the comparison systems focus narrowly on logic synthesis, using other existing algorithms (especially from SIS) for Boolean factoring, regression testing, and preliminary reduction. (Losp is a stand-alone system.) In particular, the best reported results are for HANNIBAL, aftrer the cirucit has been preprocessed by the SIS minimization tool. This is so, since most algorithms exhibit high variance across circuit types. For example, BDDs are exponential in size for arithmetic circuits, while HANNIBAL's "recursive learning" (nothing ot do with machine learning) works best after two-level circuits are optimized by SIS into multilevel versions. SIS itself performs poorly on all benchmarks, relative to the other comparison systems.

Losp Results 10-99

Circuit optimization naturally has many objectives, including layout area, delay times, required wiring, fault-tolerance and testability, and power consumption. A critical distinction is made between technology-independent optimization and technology-dependent optimization, the latter often being quite specific to the intended function library and silicon manufacturing technique. All studies reported here are technology-independent.

Appendix II includes basic descriptive and performance information. The ISCAS'91 names for each comparison circuit are followed by a brief description of the functionality of the circuit. Each circuit is characterized by its number of inputs and outputs. The raw gate count is the number of simple gates, excluding inverters, in the unprocessed netlist. The raw literal count is the number of literals in the unprocessed netlist. In a multilevel network, the literal count is the sum of mentions of variable names over all nodes.

All numerical results are literal counts for the entire circuit. The data columns identify which comparison system had the best performance on each benchmark, and the Losp performance, including the specific run in which this performance was demonstrated.

The Losp data was collected from five runs over the last year. All that changed over the different runs was the sequencing of particular reduction operations. (For example, abstracting EQUALS may defeat some reductions which require the EQUALS form to be expressed as components rather than as a single abstraction. However, not extracting EQUALS may result in an exponential explosion of literals as the circuit is processed.)

Time Performance

Algorithm timing is not included in these results. The prototyping Losp code is inefficient in several areas. Given the non-customization of the code, however, Losp processing times are all comparable to the best published times, while avoiding exponential run-away times which afflict some of the comparison systems. Losp was designed for better minimization, but does include several level-of-effort parameters to control processing time. Maximal reduction effort is necessary to be competitive. So the Losp design emphasizes better minimization performance while holding processing times comparable.

One important point is that global minimization requires exponential effort, so that in all practical systems, heuristics are necessary. Due to the much simpler representation in Losp, Losp heuristics are both powerful and inexpensive. Another point is that the extreme diversity of the benchmark circuits (ALU, DSP, logic, arithmetic, etc.), results in all systems exhibiting very high variance in processing times, ranging from a few seconds to several hours, depending on the type, and not particularly on the size.

Discussion

From Appendix II, it is easy to see that HANNIBAL with a preprocess of SIS gives the best comparison results, particularly on the larger circuits (18/34 records). HANNIBAL alone contributes another 5 records, while MIRACLE contributes 9 and CLS, the remaining 2.

Losp is able to outperform the comparison systems on 17 of the 34 circuits. The performance records then distribute as follows:

LOSP	17
HANNIBAL+SIS	13
HANNIBAL	2
MIRACLE	2
CLS	0

The approximate distribution of Losp improvements is:

N	*category*	*name*
4	< -50%	alu2, term1, dalu, alu4
2	-25% to -50%	vda, ttt2
6	-10% to -25%	9symml, c432, x4, apex6, x1, k2
5	0 to -10%	apex7, i7, x3, frg2, pair
1	equal	rot
8	0 to 10%	cht, example2, i6, i9, c3540, i8, c5315, c7552
4	10 to 25%	c8, c880, c1908, c2670
3	25% -50%	c499, c1355, c6288
1	> 50%	t481

We can see that Losp has difficulty with ALU-type circuits, while it excels at arithmetic circuits. Arithmetic circuits are known to be the most difficult for other systems. Several avenues for improving the Losp reduction techniques remain, holding hope for further performance improvements.

011291

Appendix III shows results for a collection of industrial circuits.

020529

Appendix IV shows a comparison of ILOC performance to that of the Synopsys tool. ***Appendix V*** shows the same comparison set as Appendix I, with improvements incorporated. Note: the comparison performance of other systems has not been updated.

APPENDIX I

Results under "other's record" that are enclosed in square brackets are worse than Losp performance. Double square brackets indicate recent Losp improvements

circuit	others' record	bestlosp
9symml	178	214
cht	[208]	149
c8	[150]	97
apex7	[224]	202
c432	165	227
example2	[343]	273
alu2	274	430
vda	566	719
ttt2	148	176
i6	[494]	416
c499	[[360]]	346
c880	[400]	353
i9	[[756]]	703
i7	[616]	573
term1	131	203
x4	[357]	334
c1908	[511]	394
apex6	687	718
c1355	[[360]]	346
x1	287	291
rot	[641]	608
k2	1179	1321
c2670	[[701]]	608
t481	[697]	54
x3	[758]	679
c3540	[[1144]]	1068
i8	[1195]	1195
frg2	[834]	785
pair	1509	1586
dalu	735	1497
c5315	[1679]	1561
alu4	596	1716
c7552	[1778]	1751
c6288	[[3210]]	2562

APPENDIX II

circuit	type	input/output		raw gates	raw literals
9symml	count ones	9	1	141	363
cht	logic	47	36	156	494
c8	logic	28	18	151	465
apex7	logic	49	37	163	449
c432	decoder	36	7	162	368
example2	logic	85	66	208	549
alu2	alu	10	6	205	866
vda	logic	17	39	124	1372
ttt2	logic	24	21	255	891
i6	logic	138	67	270	963
c499	error correct	41	32	370	784
c880	alu+control	60	26	306	625
i9	logic	88	63	318	1385
i7	logic	199	67	332	1237
term1	logic	34	10	410	1188
x4	logic	94	71	388	1221
c1908	error correct	33	25	345	769
apex6	logic	135	99	549	1214
c1355	error correct	41	32	506	992
x1	logic	51	35	326	2411
rot	logic	135	107	492	1742
k2	logic	45	45	249	3078
c2670	alu+control	233	140	584	1251
t481	logic	16	1	1043	2641
x3	logic	135	99	884	2325
c3540	alu+control	50	22	794	1829
i8	logic	133	81	911	4201
frg2	logic	143	139	1146	3236
pair	logic	173	137	1387	3178
dalu	dedicated alu	75	16	1476	3293
c5315	alu+selector	178	123	1324	3085
alu4	alu	14	8	2367	6582
c7552	alu+control	207	108	1894	4004
c6288	16bit multiply	32	32	2338	4674

who	others-best		Losp	run	circuit
hannibal+SIS		178	214	990206	9symml
miracle	[208]	*185		990830	cht
miracle	[150]	*114		990907	c8
hannibal+SIS		224	237	990716	apex7
hannibal+SIS		165	210	990830	c432
miracle	[343]	*338		990907	example2
hannibal		274	430	990206	alu2
hannibal+SIS		566	719	990907	vda
hannibal		148	192	990206	ttt2
miracle	[494]	*483		990206	i6
CLS+SIS	[360]	*272		981205	c499
hannibal	[400]	*353		990206	c880
miracle	[756]	*721		990206	i9
miracle		616	642	990716	i7
hannibal+SIS		131	250	990716	term1
hannibal+SIS		357	406	981205	x4
hannibal	[511]	*391		981205	c1908
hannibal+SIS		687	798	990830	apex6
CLS+SIS	[360]	*272		981205	c1355
hannibal+SIS		287	327	990907	x1
hannibal+SIS	[641]	*641		990206	rot
miracle		1179	1321	990907	k2
hannibal	[701]	*603		981205	c2670
miracle	[697]	*58		990206	t481
hannibal+SIS		758	785	990814	x3
hannibal+SIS	[1144]	*1068		990206	c3540
miracle	[1195]	*1187		981205	i8
hannibal+SIS		834	913	990206	frg2
hannibal+SIS		1509	1650	990206	pair
hannibal+SIS		735	1473	981205	dalu
hannibal+SIS	[1679]	*1561		990830	c5315
hannibal+SIS		596	1949	990716	alu4
hannibal+SIS	[1778]	*1751		990206	c7552
hannibal+SIS	[3210]	*2413		981205	c6288

APPENDIX III 011291 (TECHMAP-PARAMETERS NOR 16 6 NIL NIL)

ILOC Optimization of Industrial Circuits: NOR GATES

Name	*Brief Type*	*Before Gates*		*After Gates*	*Before Lits*	*After Lits*
bi01	FSM	42		92		46
bi02	FSM	26		53		29
bi03	arbiter	141		300		147
bi04	min/max	578	624	1166	1168	591
bi05	memory content	898	1398	1912	2781	901
bi06	interrupts	52		110		56
bi07	count points	411	637	840	1260	414
bi08	inclusions	154		326		165
bi09	converter	157		321		160
bi10	voting	172		372		185
bi11	scramble	468	1092	997	2191	477
bi12	game	1025	1166	2138	2384	1032
bi13	interface	319		643		331
bi14	Viper	4689		10051		4723
bi15	80386	8787		19446		8825
bi17	3 80386s	24060		53759		24099
bi18	8 processor	68626		149550		68655
bi20	2 Vipers	9365		20022		9399
bi21	2 Vipers	9749		20767		9783
bi22	3 Vipers	15017		31986		15051

(BI04S ((I-O 13-08) (CELL 624) (LITS 1168) (NETS 637) (PATH SEQ)) ((INV 165) (OR 0) (AND 0) (NOR 325) (NAND 0) (EQ 11) (XOR 0) (ITE 57) (REG 66) (LIB 0) (WIRE 0) (MIX 0) (GATES 393)))

(BI05S ((I-O 03-36) (CELL 1398) (LITS 2781) (NETS 1401) (PATH SEQ)) ((INV 306) (OR 0) (AND 0) (NOR 893) (NAND 0) (EQ 26) (XOR 0) (ITE 140) (REG 33) (LIB 0) (WIRE 0) (MIX 0) (GATES 1059)))

(BI07S ((I-O 03-08) (CELL 637) (LITS 1260) (NETS 640) (PATH SEQ)) ((INV 162) (OR 0) (AND 0) (NOR 349) (NAND 0) (EQ 11) (XOR 0) (ITE 66) (REG 49) (LIB 0) (WIRE 0) (MIX 0) (GATES 426)))

(BI11S ((I-O 09-06) (CELL 1092) (LITS 2191) (NETS 1101) (PATH SEQ)) ((INV 207) (OR 0) (AND 0) (NOR 774) (NAND 0) (EQ 9) (XOR 0) (ITE 71) (REG 31) (LIB 0) (WIRE 0) (MIX 0) (GATES 854)))

(BI12 ((I-O 07-06) (CELL 1166) (LITS 2384) (NETS 1173) (PATH SEQ)) ((INV 250) (OR 0) (AND 0) (NOR 587) (NAND 0) (EQ 21) (XOR 0) (ITE 187) (REG 121) (LIB 0) (WIRE 0) (MIX 0) (GATES 795)))

ILOC Optimization of Industrial Circuits AND-OR-XOR-MUX GATES

(TECHMAP-PARAMETERS AND-OR-NOT-XOR-MUX 16 6 NIL NIL)

Name	*Brief Type*	*Before Gates*		*After Gates*	*Before Lits*	*After Lits*
bi01	FSM	42		92		46
bi02	FSM	26		53		29
bi03	arbiter	141		300		147
bi04	min/max	578	748	1166	1310	591
bi05	memory content	898	2114	1912	3509	901
bi06	interrupts	52		110		56
bi07	count points	411	837	840	1472	414
bi08	inclusions	154		326		165
bi09	converter	157		321		160
bi10	voting	172		372		185
bi11	scramble	468	1696	997	2777	477
bi12	game	1025	1526	2138	2764	1032
bi13	interface	319		643		331
bi14	Viper	4689		10051		4723
bi15	80386	8787		19446		8825
bi17	3 80386s	24060		53759		24099
bi18	8 processor	68626		149550		68655
bi20	2 Vipers	9365		20022		9399
bi21	2 Vipers	9749		20767		9783
bi22	3 Vipers	15017		31986		15051

(BI04S ((I-O 13-08) (CELL 748) (LITS 1310) (NETS 761) (PATH SEQ)) ((INV 285) (OR 215) (AND 110) (NOR 0) (NAND 0) (EQ 0) (XOR 11) (ITE 57) (REG 66) (LIB 0) (WIRE 4) (MIX 0) (GATES 393)))

(BI05S ((I-O 03-36) (CELL 2114) (LITS 3509) (NETS 2117) (PATH SEQ)) ((INV 982) (OR 777) (AND 116) (NOR 0) (NAND 0) (EQ 0) (XOR 26) (ITE 140) (REG 33) (LIB 0) (WIRE 40) (MIX 0) (GATES 1059)))

(BI07S ((I-O 03-08) (CELL 837) (LITS 1472) (NETS 840) (PATH SEQ)) ((INV 343) (OR 269) (AND 89) (NOR 0) (NAND 0) (EQ 0) (XOR 11) (ITE 66) (REG 49) (LIB 0) (WIRE 10) (MIX 0) (GATES 435)))

(BI11S ((I-O 09-06) (CELL 1696) (LITS 2777) (NETS 1709) (PATH SEQ)) ((INV 797) (OR 656) (AND 105) (NOR 0) (NAND 0) (EQ 0) (XOR 9) (ITE 71) (REG 31) (LIB 0) (WIRE 25) (MIX 2) (GATES 841)))

(BI12 ((I-O 07-06) (CELL 1526) (LITS 2764) (NETS 1533) (PATH SEQ)) ((INV 586) (OR 433) (AND 156) (NOR 0) (NAND 0) (EQ 0) (XOR 21) (ITE 187) (REG 121) (LIB 0) (WIRE 22) (MIX 0) (GATES 797)))

APPENDIX IV SYNOPSIS COMPARISON

Seven circuits, EDIF format as output of SYNOPSIS DESIGN COMPILER v1998.08

ID	DESCRIPTION	I/O	VHDL lines, procs	
bi04	Compute min and max	11/8	102	1
bi05	Elaborate the contents of a memory	1/36	332	3
bi07	Count points on a straight line	1/8	92	1
bi11	Scramble string with variable cipher	7/6	118	1
bi13	Interface to meteo sensors	10/10	296	5
bi14	Vipor processor core	32/54	509	1
bi15	80386 processor core	36/70	671	3

ID	GATES	FF+INV+GATES			+I/O	GATE LIBRARY			
bi04	578	66	49	463	597	nand-3-4-5	and-3	or-3	nor
bi05	898	34	91	773	935	nand-3-4-5	and-3-4	or-3-4-5	nor-3-4-5
bi07	411	49	36	326	420	nand-3-4		or-3-4-5	nor
bi11	468	31	49	388	481	nand-3-4-5	and-3-4	or-3	nor-3
bi13	319	53	30	236	339	nand-3-4-5	and-3	or-3-4	nor-3
bi14	4689	245	320	4124	4775	nand-3-4-5	and-3-4-5	or-3-4	nor-3-4-5
bi15	8787	671	494	7844	8893	nand-3-4-5	and-3-4-5	or-3-4	nor-3-4-5

COMPARISON of LITERAL COUNT

	SYNOPSIS	low-effort		medium-effort		%reduction
bi04	1166	730	63	722	62	38
bi05	1912	960	50	960	50	50
bi07	840	563	67	563	67	33
bi11	997	677	68	652	65	35
bi13	643	436	68	416	65	35
bi14	10051	6174	61	6012	60	40
bi15	19446	12394	64	12335	63	37

Header from Synopsys EDIF results

```
(edif Synopsys_edif (edifVersion 2 0 0) (edifLevel 0)
 (keywordMap (keywordLevel 0))
 (status  (written (timeStamp 1999 6 30 20 51 17)
   (program "Synopsys Design Compiler" (Version "1998.08"))
   (dataOrigin "company") (author "designer")))
```

APPENDIX V

ILOC™ OPTIMIZATION COMPARED TO BEST IN WORLD (combinational only)

circuit	others' record	01-raw	02-inv	low-effort	mod-effort	high-effort	%world
9symml	178	372	363	236	229	214	120
cht	[208]	499	494	185			89
c8	[150]	522	465	137	123	114	76
apex7	224	495	449	235			105
c432	165	531	368	244	226	210	127
example2	[343]	574	549	360	339	338	99
alu2	274	896	866	447		430	157
vda	566	1389	1372	708	706		125
ttt2	148	968	899	200		192	130
i6	[494]	1033	963	549		483	98
c499	[360]	938	784	556	380	272	76
c880	[400]	957	633	405	362	353	88
i9	[756]	1395	1385	750		703	93
i7	616	1307	1237	701	675	642	104
term1	131	1377	1211	225			172
x4	357	1345	1232	416	406		114
c1908	[511]	1203	769	535	404	391	77
apex6	687	1287	1214	822		798	116
c1355	[360]	1426	992	556	380	272	76
x1	287	2453	2411	325			113
rot	[641]	1863	1744	702	693	641	100
k2	1179	3121	3078	1311			111
c2670	[701]	1961	1267	732	583		83
t481	[697]	2657	2641	61		58	8
x3	758	2734	2369	786		780	103
c3540	[1144]	2746	1841	1199	1148	1068	93
i8	[1195]	4325	4201	1145			96
frg2	[834]	3755	3278	937		793	95
pair	1509	4016	3196	1718	1687	1650	109
dalu	735	4458	3383	1489		1473	200
c5315	[1679]	4558	3052	1833	1516		90
alu4	596	6599	6582	1727			290
c7552	[1778]	6218	3978	2232	1659		93
c6288	[3210]	7152	4676	3312		2413	75