ISSN 2186-5647

低消費電力指向マルチサイクルキャプチャテスト生成における時間展開数の解析

日大生産工　○山崎 紘史　NEC エンジニアリング　西間木 淳
日大生産工　細川 利典　京都産大　吉村 正義

1. はじめに

近年の半導体微細化技術の進歩に伴って，大規模集積回路（Very Large Scale Integrated circuits: VLSI）が大規模化・高速化し，テスト実行時における消費電力の増大が問題となっている．テスト時の過度な消費電力は IR ドロップを発生させ,回路内に遅延を引き起こすことで良品を不良品と判断する誤テストを行う可能性がある．また，消費電力による発熱で回路を破壊する可能性も考えられる．これらの現象により，テスト時の消費電力増加は歩留まり損失の原因の一つとして考えられる．したがって，歩留まり損失を抑制するためにテスト時の消費電力を削減することが重要である．

VLSI のテスト方法として広く普及しているスキャンテスト[1]では，テスト時特有の消費電力としてスキャンチェインへのテストデータの印加を行うシフト動作時に発生するシフト電力[2]と，テスト応答のフリップフロップ(Flip-Flop : FF)への格納を行うキャプチャ動作時に発生するキャプチャ電力[2]が挙げられる．シフト動作時には最長スキャンパス長分のクロックパルスが必要となるため，シフト電力の影響は温度上昇という形で現れ，VLSI を高熱で破壊する可能性がある．一方，キャプチャ動作時には FF の遷移が同じタイミングで発生し,回路内の信号線が遷移する.このとき，過度の信号線遷移が発生すると，キャプチャ電力の影響は IR ドロップによる回路内の遅延増加という形で現れる．IR ドロップによる遅延増加は，正常な VLSI を不良と判定する誤テストを招く可能性がある．本論文では，フルスキャン設計[1]が施された順序回路の遷移故障モデルを対象とするテスト生成において，キャプチャ電力の削減を試みる．

遷移故障のテスト方法として，2 パターンテストが提案されている[2]．また 2 パターンテストを用いた実速度スキャンテスト法として，スキュードロード方式[3][4]とブロードサイド方式[5]が提案されている．本論文ではブロードサイド方式を用いる．キャプチャ時高消費電力の原因として,VLSI の機能動作を考慮せず,構造的なテスト生成を行っていることが考えられる．フルスキャン設計が施された回路において，一般的な自動テスト生成(Automatic Test Pattern Generation: ATPG)は機能動作を考慮せずテスト生成の容易性を優先してテスト生成を行う．そのため，生成されたテストパターンのスキャン FF の状態が機能動作では起こりえない状態となり，回路内の多くの信号線に遷移を発生させている可能性がある．

一方，文献[6]においてブロードサイドテスト用のテストパターンに対してマルチサイクル間(5 サイクル以上)キャプチャ動作を実行することでキャプチャ電力が減少することが報告されている． また，文献[7]において，k(≧2)時間展開モデルを利用して低消費電力な遷移故障のテスト生成をおこなう低消費電力指向マルチサイクルキャプチャテスト生成が提案されている．しかしながら，文献[6][7]において，キャプチャ時消費電力の削減に効果的な時間展開数の解析は行われていない．本論文では，文献[7]で報告されている低消費電力指向マルチサイクルキャプチャテスト生成に対して,より低消費電力なテストパターンが生成可能な時間展開数を解析し，評価を行う．

2 章で低消費電力指向マルチサイクルキャプチャテスト生成について説明し，3 章で低消費電力指向マルチサイクルキャプチャテスト生成の時間展開数の解析,4 章で実験結果について考察し,5 章で結論と今後の課題について述べる．

2. 低消費電力指向マルチサイクルキャプチャテスト生成

本章では，文献[7]で提案された低消費電力指向マルチサイクルキャプチャテスト生成について説明する．

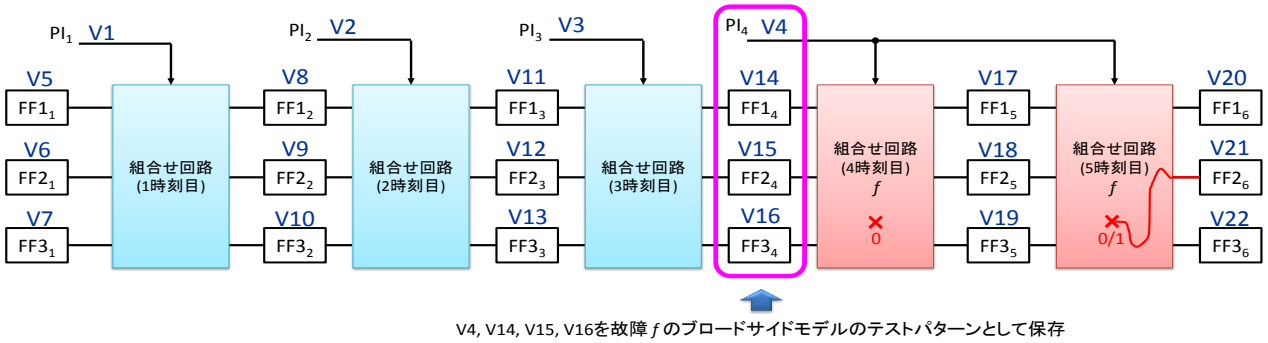
2.1. 低消費電力指向マルチサイクルキャプチャテスト生成のテスト生成モデル

文献[7]において，k(≧2)時間展開モデルを利用して低消費電力な遷移故障のテスト生成をおこなう低消費電力指向マルチサイクルキャプチャテスト生成が提案されている．文献[7]の低消費電力指向マルチサイクルキャプチャテスト生成では，k(≧2)サイクル間キャプチャ動作を行って最終時刻で遷移故障を励起･検出し，k-1 時刻におけるスキャン FF と外部入力(Primary Inputs: PI)の値をテストパターンとする．また，テストはブロードサイド方式で実行する．

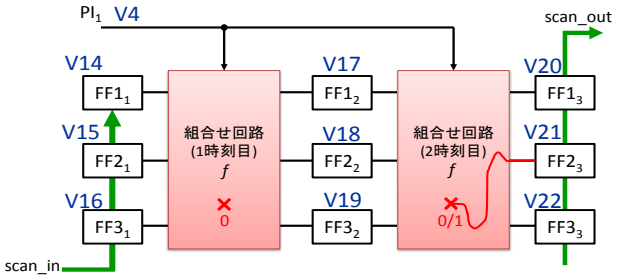
図 1(a)に低消費電力指向マルチサイクルキャプチャテスト生成のモデル例として，信号線 f の立上り遷移故障に対する 5 時間展開モデル(k=5)を示す．図 1(a)において，1 時刻目の FF である $FF1_1$，$FF2_1$，$FF3_1$ の 3 個の FF は可制御である．また最終時刻+1 の FF である $FF1_6$，$FF2_6$ ，$FF3_6$ の 3 つの FF は可観測である．また，PI_1，PI_2，PI_3，PI_4 の各外部入力も可制御であり，4 時刻目と 5 時刻目の外部入力である PI_4 のみ同一の値が入力される．また 4 時刻目の組合せ回路部には，信号線 f に立上り遷移故障の初期値である 0 を割当てる．

An Analysis of Time Expansion Size for Low Power Oriented Multi Cycle Capture Test Generation

Hiroshi YAMAZAKI, Jun NISHIMAKI, Toshinori HOSOKAWA, and Masayoshi YOSHIMURA



(a) 低消費電力指向マルチサイクルキャプチャテスト生成のテスト生成モデル例(k=5)



(b)低消費電力指向マルチサイクルキャプチャテスト生成のテスト実行例

図 1. 低消費電力指向マルチサイクルキャプチャテスト生成

5 時刻目の組合せ回路部には，信号線 f に立上り遷移故障の故障値である 0/1 を割当てる．1 時刻目から 3 時刻目の組合せ回路部に関しては，故障は仮定しない．このようなテスト生成モデルを用いてテスト生成を行い，各 PI と各 FF に対する論理値 V1～V22 を求める．テスト生成後，PI_4 と $FF1_4$，$FF2_4$，$FF3_4$ に割当てられた論理値 V4，V14，V15，V16 を故障 f の立上り遷移故障のブロードサイド方式のテストパターンとして保存する．このようなテスト生成モデルを用いることで，キャプチャ動作を複数サイクル間実行した後の FF の状態をテストパターンとするため，キャプチャ電力が抑制されたブロードサイド方式のテストパターンが得られる．

低消費電力指向マルチサイクルキャプチャテスト生成では，テストはブロードサイド方式を用いる．図 1(b)に，図 1(a)の低消費電力指向マルチサイクルキャプチャテスト生成で生成したテストパターンに対するテスト実行例を示す． 図 1(a)では，信号線 f の立上り遷移故障のテストパターンとして，PI_4=V4，$FF1_4$=V14，$FF2_4$=V15，$FF3_4$=V16 を保存した．テスト時には，図 1(b)に示すように，外部入力 PI_1 に V4 を印可する．また，スキャンイン端子 scan_in より，$FF1_1$=V14，$FF2_1$=V15，$FF3_1$=V16 を印可する．その後，2 サイクルキャプチャ動作を行い，故障 f の故障影響を $FF2_3$ に伝搬させる．故障影響の観測はスキャンアウト端子 scan_out より，スキャン FF に対してシフト動作を行うことで観測する．

2.2. 低消費電力指向マルチサイクルキャプチャテスト生成アルゴリズム

本節では，低消費電力指向マルチサイクルキャプチャテスト生成のアルゴリズムを示す．図 2 に低消費電力指向マルチサイクルキャプチャテスト生成の全体アルゴリズムを示す．

回路 C と時間展開数 k，高消費電力テストパターンを判定するための最も重み付き信号遷移(Weighted Switching Activity: WSA) [8]閾値 th を入力とする．まず高消費電力テスト集合 HPT と低消費電力テスト集合 LPT を Φに初期化する(行 4，行 5)．次にテスト生成の対象となる全ての遷移故障を算出し故障集合 F に代入する(行 6)．行 6 で算出した F と C を基に，ブロードサイドモデルでテスト生成を行い，テスト集合 T を算出する(7 行)．T に対してブロードサイドモデルで故障シミュレーションを実行し，検出故障集合 D を算出する(行 8)．T に対してドントケア判定[9]を実行し，ドントケアを含むテスト集合 XT を生成する(行 9)．XT に対してキャプチャ時消費電力を低減するドントケア割当て[10]を実行し，テスト集合 FT を生成する(行 10)．FT に含まれる各テストパターン ft_i に対して行 12 から行 19 の処理を適用する．ft_i に対して WSA を算出し wsa_i を求める(行 12)．wsa_i が th より大きいか否かを判

```
1.  Procedure  low_power_multi_cycle_test_generation( C , k , th );
2.   C : circuit,    k : time_expansion_size,    th : wsa_threshold
3.  {
4.          HPT = Φ ;
5.          LPT = Φ ;
6.          F = collect_all_fault( C );
7.          T = broadside_test_generation( C , F );
8.          D = broadside_fault_simulation( C , T );
9.          XT = x_identification( C , D , T );
10.         FT = low_power_x_filling( C , XT );
11.         for each test pattern ft_i in FT {
12.                 wsa_i = calc_wsa( C , ft_i );
13.                 if( wsa_i > th ){
14.                         HPT = HPT ∪ ft_i ;
15.                 }
16.                 else{
17.                         LPT = LPT ∪ ft_i ;
18.                 {
19.         }
20.         HPF = broadside_fault_simulation( C , HPT, F ):
21.         LPF = broadside_fault_simulation( C , LPT , F );
22.         HPF = HPF - LPF ;
23.         for each fault hpf_i in HPF {
24.                 mt_i = multi_cycle_capture_test_generation( C , k , hpf_i );
25.                 LPT = LPT∪ mt_i ;
26.                 DM = broadside_fault_simulation( C , mt_i , HPF);
27.                 HPF = HPF - DM ;
28.         }
29.         return ( LPT);
30. }
```

図 2. 低消費電力指向マルチサイクルキャプチャテスト生成の全体アルゴリズム

定する(行 13). もし wsa_i が th より大きければ ft_i を HPT に加え，HPT を更新する(行 14). それ以外の場合は，ft_i を LPT に加え，LPT を更新する (行 17). HPT に対してブロードサイドモデルで故障シミュレーションを実行し，検出故障 HPF を算出する(行 20). LPT に対してブロードサイドモデルで故障シミュレーションを実行し，検出故障 LPF を算出する(行 21). HPF から LPF の差集合をとり，HPT でしか検出できないアンセーフ故障集合 HPF を更新する(行 22). HPF に含まれる各故障 hpf_i に対して行 24 から行 28 の処理を適用する(行 23). hpf_i に対して時間展開数 k でマルチサイクルキャプチャテスト生成モデルを用いた低消費電力指向テスト生成を行い，th より WSA が低いテストパターン mt_i を生成する(行 24). mt_i を LPT に加え，LPT を更新する(行 25). mt_i に対してブロードサイドモデルで故障シミュレーションを実行し，検出故障集合 DM を算出する(行 26). HPF から DM の差集合をとり，HPF を更新する(行 27). テスト集合 LPT を返す(行 29).

3. 低消費電力指向マルチサイクルキャプチャテスト生成における時間展開数の解析

文献[6][7]において，マルチサイクル動作に着目した低消費電力なテストパターン生成法が提案されているが，時間展開数に関しては考慮されていない．そのため，低消費電力なテストパターン(セーフパターン[7])の生成に適さない時間展開数を用いている可能性がある．本章では，シフト動作によりランダムパターンを印加後，キャプチャ動作を 50 サイクル間実行させ，各サイクルのセーフパターン数からセーフパターン生成に効果的な時間展開数を解析する．また，2 サイクル目以降の外部入力の値はランダムパターンを印加する．低消費電力の閾値は，遷移故障に対してテスト生成を行い，生成されたテストパターン中の WSA の高い値の 50%，60%，70%，80%とした．各閾値以下の WSA を持つテストパターンをセーフパターンとする．

図 3 に，s9234 に対するランダムパターンを用いたマルチサイクルキャプチャ動作とセーフパターン数の解析結果を示す．解析に使用したランダムパターン数は 10,000 である．図 3 において，縦軸はセーフパターン数を示し，横軸はキャプチャサイクル数を示す.「閾値 50 以下」は遷移故障テスト集合の最大 WSA の 50%以下を閾値とした場合のセーフパターン数を示す.「閾値 60 以下」,「閾値 70 以下」,「閾値 80 以下」も同様に，遷移故障テスト集合の最大 WSA の 60%以下，70%以下，80%以下を閾値とした場合のセーフパターン数を

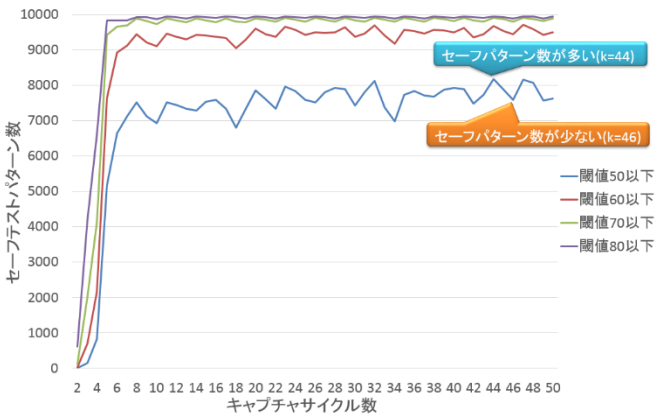


図 3. s9234 に対するマルチサイクルキャプチャ動作とセーフパターン数の解析

表 I. s9234 に対するセーフパターン数の多いキャプチャサイクル数 Top5

WSA閾値	セーフパターン数の多いキャプチャサイクル数(k				
	1位	2位	3位	4位	5位
閾値50以下	44	47	32	48	23
閾値60以下	47	32	44	23	29
閾値70以下	47	23	41	44	35
閾値80以下	35	38	47	41	11

示す．また，「閾値 50 以下」，「閾値 60 以下」，「閾値 70 以下」,「閾値 80 以下」の閾値 WSA はそれぞれ 1256，1507，1759，2010 である．

図 3 において，4 サイクル以下のマルチサイクルキャプチャ動作では，セーフパターン数が極端に少ないことがわかる．また，5 サイクル以上のマルチサイクルキャプチャ動作であっても，各サイクルでセーフパターン数に差があることがわかる.閾値 50 以下に着目すると，キャプチャサイクル数 46(k=46)より，44(k=44)のほうがセーフパターン数が多い．また，低消費電力閾値を 50%，60%，70%，80%と増加させていくと，セーフパターン数が多くなることがわかる．

図 3 の解析結果から，キャプチャサイクル数を増加させるだけではセーフパターン数が多くなるとは限らない．また，セーフテストパターン数が多く生成できるキャプチャサイクル数と，少なく生成されるキャプチャサイクル数が存在することがわかった．他のベンチマーク回路に関しても同様の結果が得られた．

表 I に，図 3 に対するセーフパターン数の多いキャプチャサイクル数の上位 5 つを示す．表 I より，各閾値に対して同一のキャプチャサイクル数が同じ順位になるとは限らない．よって，本論文では各閾値におけるセーフパターン数の多いキャプチャサイクル数の上位 5 つから，最も出現頻度の高いキャプチャサイクル数を 5 つ選出し，低消費電力指向マルチサイクルキャプチャテスト生成の時間展開数(k)に利用する．

4. 実験結果

本章では ISCAS’89 ベンチマーク回路に対する実験結果を示す．故障モデルは遷移故障である．時間展開数 k は文献[7]と同じ 2,5,10,15,20 と，3 章で示した時間展開数の解析結果を利用した 2 種類で実験を行っている．また，k=2 は従来のブロードサイド方式のテスト生成と同一である．

表 II に図 2 の行 22 までの処理を適用した初期テスト生成結果を示す．表 II において，「閾値割合」はアンセーフパターン[7]を判定するための閾値 WSA を計算するための割合を示す．閾値割合は最高 WSA の 70%と 80%を用いた．「WSA 閾値」はアンセーフパターン判定の WSA 閾値を示し，生成されたテスト集合の最高 WSA の 70%と 80%から算出した.「アンセーフパターン数」は WSA 閾値を超えたテストパターン数，「アンセーフ故障数」はアンセーフパターンでのみ検出可能な故障数を示す．

表 III に表 II のアンセーフ故障を対象に，図 2 の行 23 から 30 までの処理を適用した結果を示す．表 III において「対象故障数」は表 II でアンセーフ故障と判定された故障を示し，表 II の「アンセーフ故障数」と同数である．表 III の「アンセーフ故障数」は表 II で判定されたアンセーフ故障を対象に，再度低消費電力指向マルチサイクルキャプチャテスト生成を行い，その結果アンセーフ故障と判定された数を示す.「テスト不可能故障判定数」は k 時間展開テスト生成モデルにおいてテスト不可能と判定された故障数を示す.また，

「k=2」は時間展開数 2 での低消費電力指向マルチサイクルキャプチャテスト生成を示し,他も同様である.また,「LOC」は k=2 のテスト生成結果を示し,「文献[7]」は文献[7]と同じ時間展開数(k=5, 10, 15, 20)のテスト生成結果を示す.「提案手法」は,3 章で示した時間展開数の解析結果を利用した時間展開数でのテスト生成の結果を示す.また,「1 位」から「5 位」は 3 章で示したセーフパターン数の多いキャプチャサイクル数の上位 5 つの順位を示す.「提案手法」に関しては,時間展開数の解析によって得られる時間展開数が回路によって異なる.そのため,回路ごとで時間展開数が異なる.

「文献[7]」と「提案手法」のアンセーフ故障数を比較すると,s13207 の閾値 80%と s38417 の閾値 70%を除いた全ての回路に対して「提案手法」のほうがアンセーフ故障数が少ない結果となった.また,「文献[7]」のアンセーフ故障数の平均は 473,「提案手法」のアンセーフ故障数の平均は 366 であり,各回路の平均においても少ない結果となった.テスト不可能故障判定数では,「文献[7]」と「提案手法」に大きな差が無い結果となった.

5. まとめ

本論文では,低消費電力指向マルチサイクルキャプチャテスト生成におけるセーフパターン数に効果的な時間展開数の解析をおこなった.提案手法により,従来の低消費電力指向マルチサイクルキャプチャテスト生成よりアンセーフ故障数の削減ができた.また,解析結果で得られた時間展開数のうち「1 位」の時間展開数が比較的多くのアンセーフ故障を削減できる傾向があることがわかった.今後の課題として,アンセーフ故障数のさらなる削減が挙げられる.

文献

[1] H. Fujiwara, “Logic Testing and Design for Testability,” The MIT Press, (1985) pp298.

[2] A.Krstic, and K-T Cheng, “Delay Fault Testing for VLSI Circuits,” Kluwer Academic Publishers, 1998.

[3] J.Savir. “Skewd-Load Transition Test: Part 1:, Calculus.” Proceedings of IEEE International Test Conference, pp705-713, 1992

[4] J.Savir. “Skewd-Load Transition Test: Part 2:, Calculus. ”Proceedings of IEEE International Test Conference, pp714-722, 1992

[5] Xiao Liu and Michael S. Hsiao “Constrained ATPG for Broadside Transition Testing” Department of Electrical & Computer Engineering, Virginia Tech, Blacksburg, VA 24061.

[6] E. K. Moghaddam, J. Rajski, S. M. Reddy, M. Kassab, “At-Speed Scan Test with Low Switching Activity,” IEEE VLSI Test Symposium, pp177-182, 2010.

[7] “A Multi Cycle Capture Test Generation Method for Low Capture Power Dissipation”, Hiroshi Yamazaki, Jun Nishimaki, Toshinori Hosokawa, and Masayoshi Yoshimura, Design Automation & Test in Europe 2015 Designing with Uncertainty Opportunities & Challenges, pp21-23, 2015.

[8] Sying-Jyan Wang, Kuo-Lin Fu, Katherine Shu-Min Li, “Low Peak Power ATPG for n-Detection Test”, p1, 1999.

[9] K. Miyase, K. Noda, H. Ito, K. Hatayama, T. Aikyo, Y. Yamato, H. Furukawa, X. Wen and S. Kajihara, “Effective IR-Drop Reduction in At-Speed Scan Testing Using Distribution-Controlling X-Identification,” IEEE/ACM International Conference on Computer-Aided Design, pp. 52-58, 2008.

[10] Xiaoqing Wen,Kohei Miyase,Seiji Kajihara,Tatsuya Suzuki,Yuta Yamato,Patrick Girard,Yuji Ohsumi,Laung-Terng Wang, "A Novel Scheme to Reduce Power Supply Noise for High-Quality At-Speed Scan Testing," International Test Conference, Paper 25.1, 2007.

表 II. 初期テスト生成結果

回路名	閾値割合	対象故障数	検出故障数	テスト不可能故障数	打ち切り故障数	テストパターン数	WSA閾値	アンセーフパターン数	アンセーフ故障数
s5378	70%	10590	6546	4044	0	223	1111	12	156
	80%						1270	6	71
s9234	70%	18468	13813	4655	0	581	1759	71	519
	80%						2010	32	210
s13207	70%	26358	19261	7097	0	627	1919	52	372
	80%						2193	12	85
s15850	70%	31694	20009	11685	0	498	2252	8	120
	80%						2574	3	51
s35932	70%	71224	49278	21946	0	113	8740	22	4922
	80%						9988	13	2326
s38417	70%	76678	73736	2942	0	1307	8222	46	2688
	80%						9396	22	1448
s38584	70%	76864	50138	26726	0	1820	4499	13	1448
	80%						5142	7	1095

表 III. 再テスト生成後のアンセーフ故障数

回路名	閾値割合	対象故障数	アンセーフ故障数										テスト不可能故障判定数									
			LOC	文献[7]				提案手法					LOC	文献[7]				提案手法				
								1位	2位	3位	4位	5位						1位	2位	3位	4位	5位
			k=2	k=5	k=10	k=15	k=20	k=34	k=40	k=29	k=39	k=35	k=2	k=5	k=10	k=15	k=20	k=34	k=40	k=29	k=39	k=35
s5378	70%	156	156	61	51	64	58	39	37	46	41	66	0	29	47	56	56	56	56	56	56	56
	80%	71	71	19	17	40	18	12	14	16	15	39	0	11	16	16	16	16	16	16	16	16
			k=2	k=5	k=10	k=15	k=20	k=47	k=32	k=44	k=23	k=35	k=2	k=5	k=10	k=15	k=20	k=47	k=32	k=44	k=23	k=35
s9234	70%	519	516	93	58	40	56	53	33	48	43	37	0	296	297	297	298	299	299	299	299	299
	80%	210	206	14	23	8	25	22	4	3	10	31	0	126	127	127	127	127	127	127	127	127
			k=2	k=5	k=10	k=15	k=20	k=46	k=45	k=37	k=23	k=49	k=2	k=5	k=10	k=15	k=20	k=46	k=45	k=37	k=23	k=49
s13207	70%	372	372	257	107	110	119	94	99	95	113	97	0	39	140	140	140	140	140	140	140	140
	80%	85	85	46	22	17	29	17	22	22	17	19	0	18	36	36	36	36	36	36	36	36
			k=2	k=5	k=10	k=15	k=20	k=24	k=28	k=32	k=40	k=44	k=2	k=5	k=10	k=15	k=20	k=24	k=28	k=32	k=40	k=44
s15850	70%	120	119	27	15	18	0	9	3	51	13	11	0	19	25	25	26	26	26	26	26	26
	80%	51	51	3	0	0	0	0	1	0	0	0	0	11	12	12	12	12	12	12	12	12
			k=2	k=5	k=10	k=15	k=20	k=39	k=40	k=44	k=22	k=21	k=2	k=5	k=10	k=15	k=20	k=39	k=40	k=44	k=22	k=21
s35932	70%	4922	4922	4363	2962	2460	2218	3140	2511	2492	1950	2494	0	0	0	0	0	0	0	0	0	0
	80%	2326	1111	2065	887	1448	946	983	933	1124	563	668	0	0	0	0	0	0	0	0	0	0
			k=2	k=5	k=10	k=15	k=20	k=22	k=21	k=27	k=29	k=24	k=2	k=5	k=10	k=15	k=20	k=22	k=21	k=27	k=29	k=24
s38417	70%	2688	2688	1094	385	139	226	255	298	338	270	361	0	36	38	38	38	38	38	38	38	38
	80%	1448	1448	253	60	64	45	43	75	47	82	69	0	19	20	20	20	20	20	20	20	20
			k=2	k=5	k=10	k=15	k=20	k=47	k=48	k=35	k=36	k=38	k=2	k=5	k=10	k=15	k=20	k=47	k=48	k=35	k=36	k=38
s38584	70%	1448	1448	804	747	725	645	738	756	638	686	484	0	31	31	31	31	31	31	31	31	31
	80%	1095	1095	577	640	709	620	493	493	507	526	354	0	15	15	15	15	15	15	15	15	15