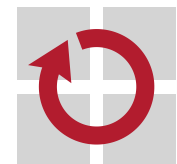
Ein Kontrollflussüberwachungsdienst für KESO-Anwendungen

Simon Schuster

03.07.2015

Department of Computer Science 4
Friedrich-Alexander-Universität
Erlangen-Nürnberg



Zuverlässigkeit

Zuverlässigkeit und Fehlertoleranz

Kontrollflussfehler

Datenfehler

Kontrollflussüberwachung

„We present a[n] [...] version [...] (ECCA) [...] targeted for the **detection of control flow-errors** with **low overhead** and **low detection latency** [...]“

- Alkhalifa et al.: Design and Evaluation of System-Level Checks for On-Line Control Flow Error Detection

Zuverlässigkeit

Zuverlässigkeit und Fehlertoleranz

Kontrollflussfehler

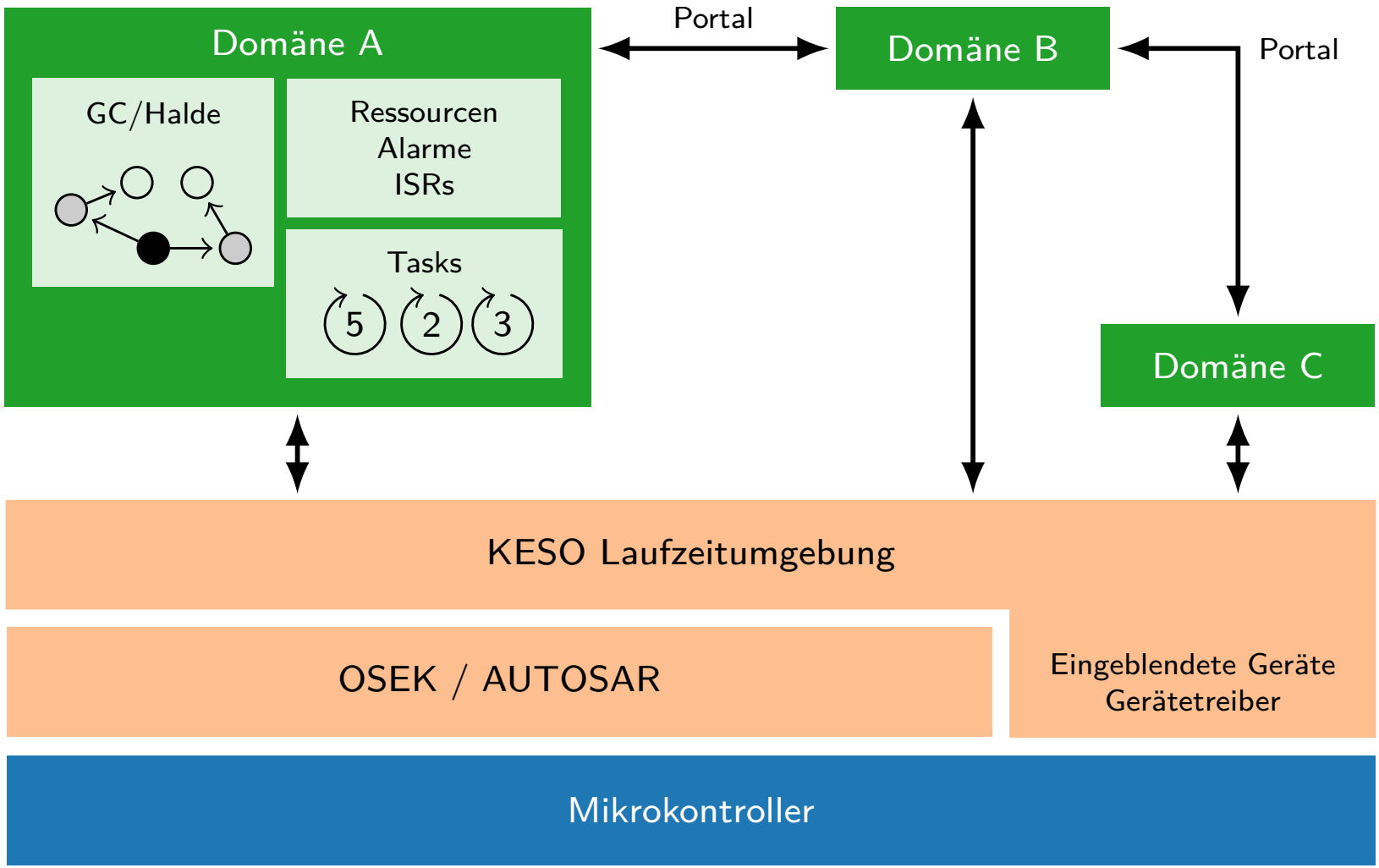
Datenfehler

Kontrollflussüberwachung

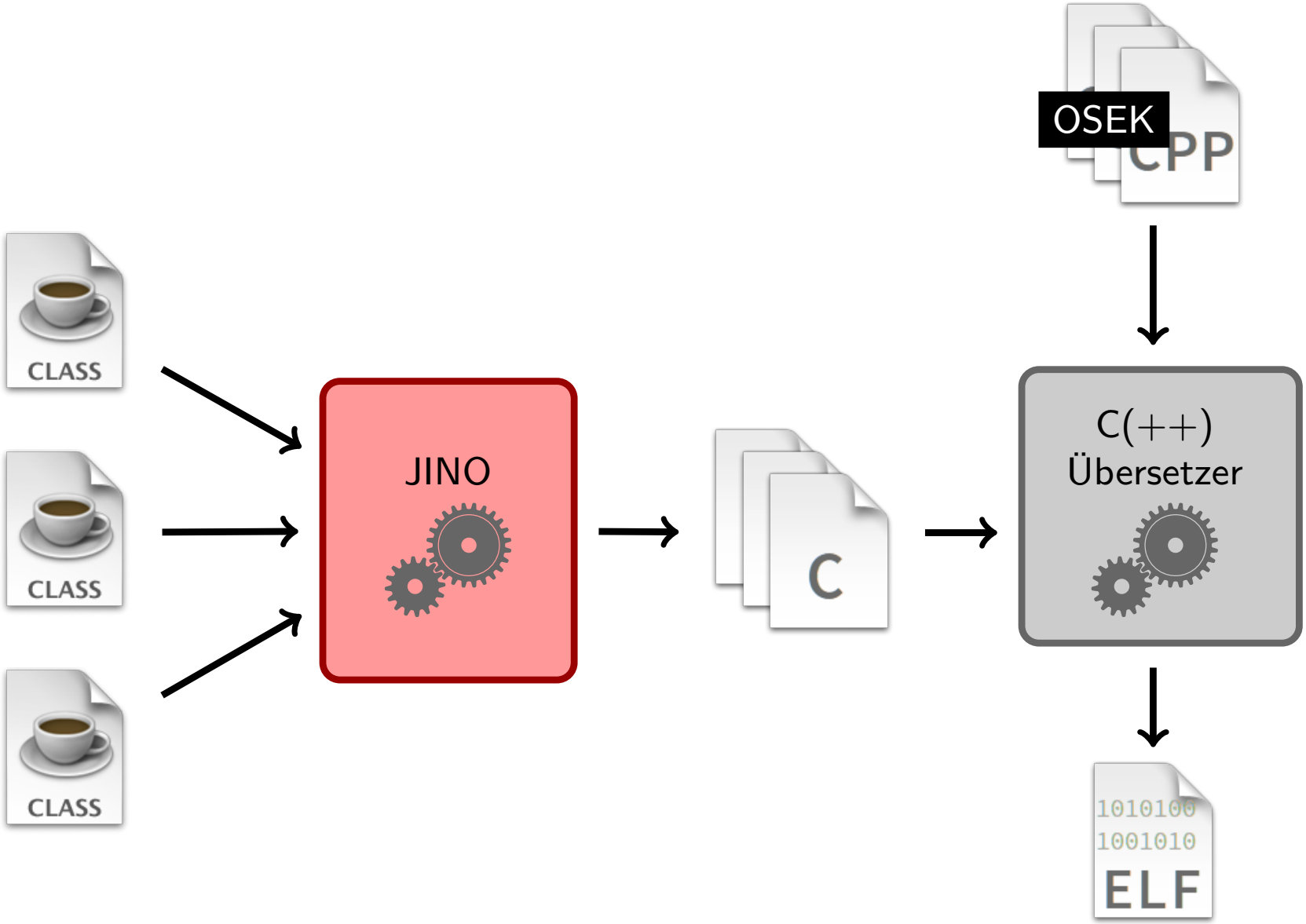
„We present a[n] [...] version [...] (ECCA) [...] targeted for the **detection of control flow-errors** with **low overhead** and **low detection latency** [...]“

- Alkhalifa et al.: Design and Evaluation of System-Level Checks for On-Line Control Flow Error Detection

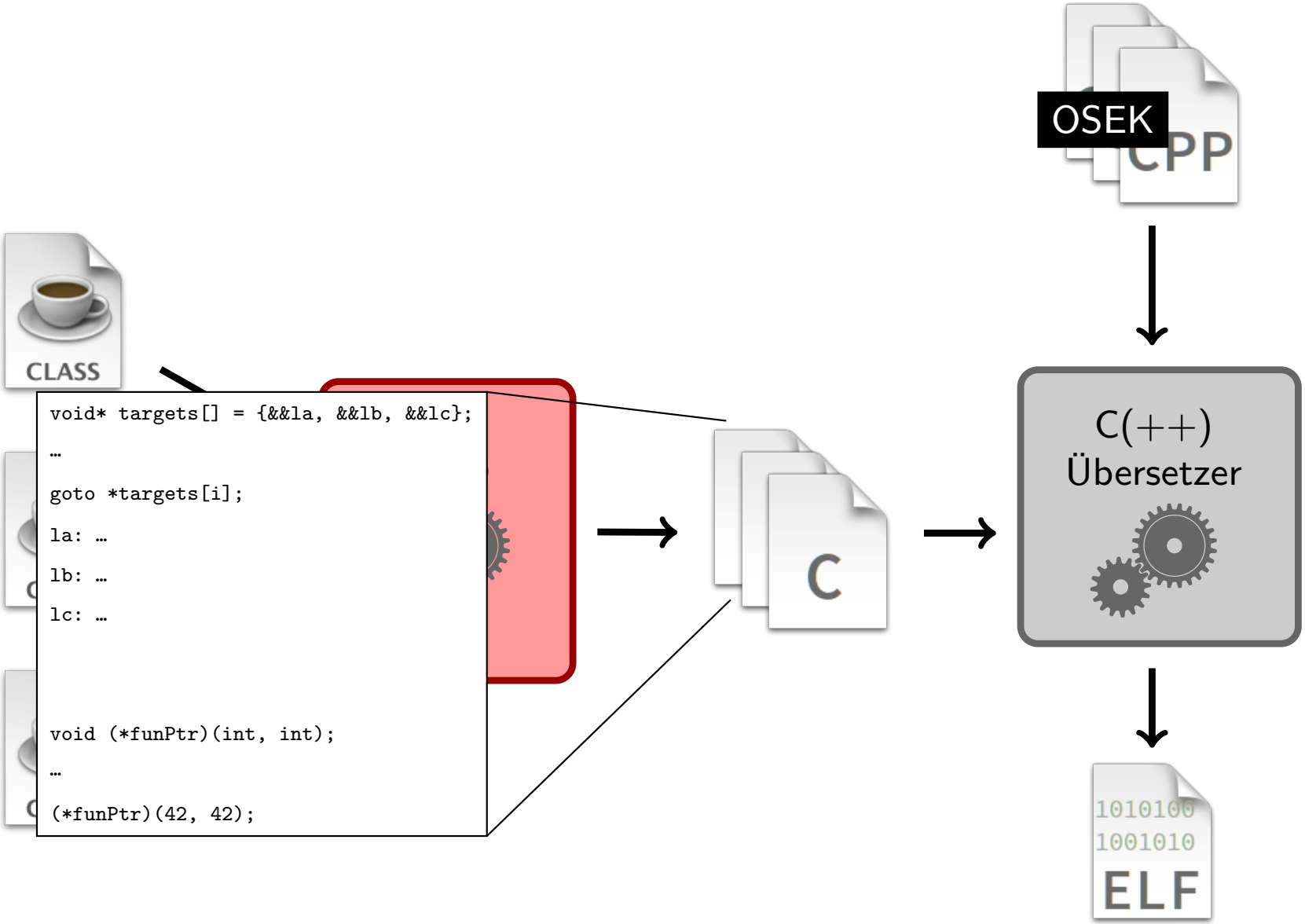
KESO-Aufbau


Domäne A
GC/Halde
Ressourcen
Alarme
ISRs
Tasks
5
2
3
Portal
Domäne B
Portal
Domäne C
KESO Laufzeitumgebung
OSEK / AUTOSAR
Eingeblendete Geräte
Gerätetreiber
Mikrokontroller

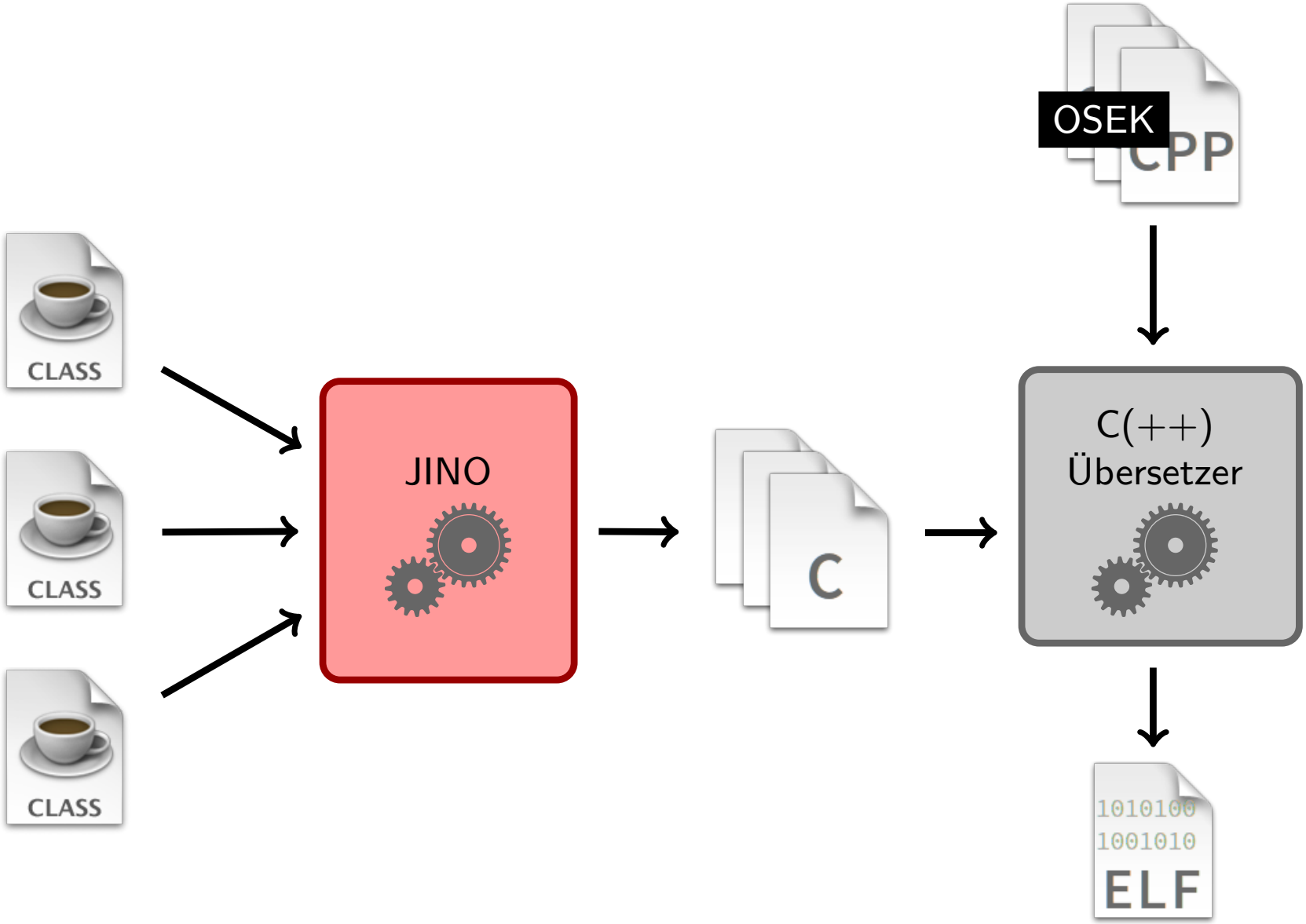
KESO-Bauprozess



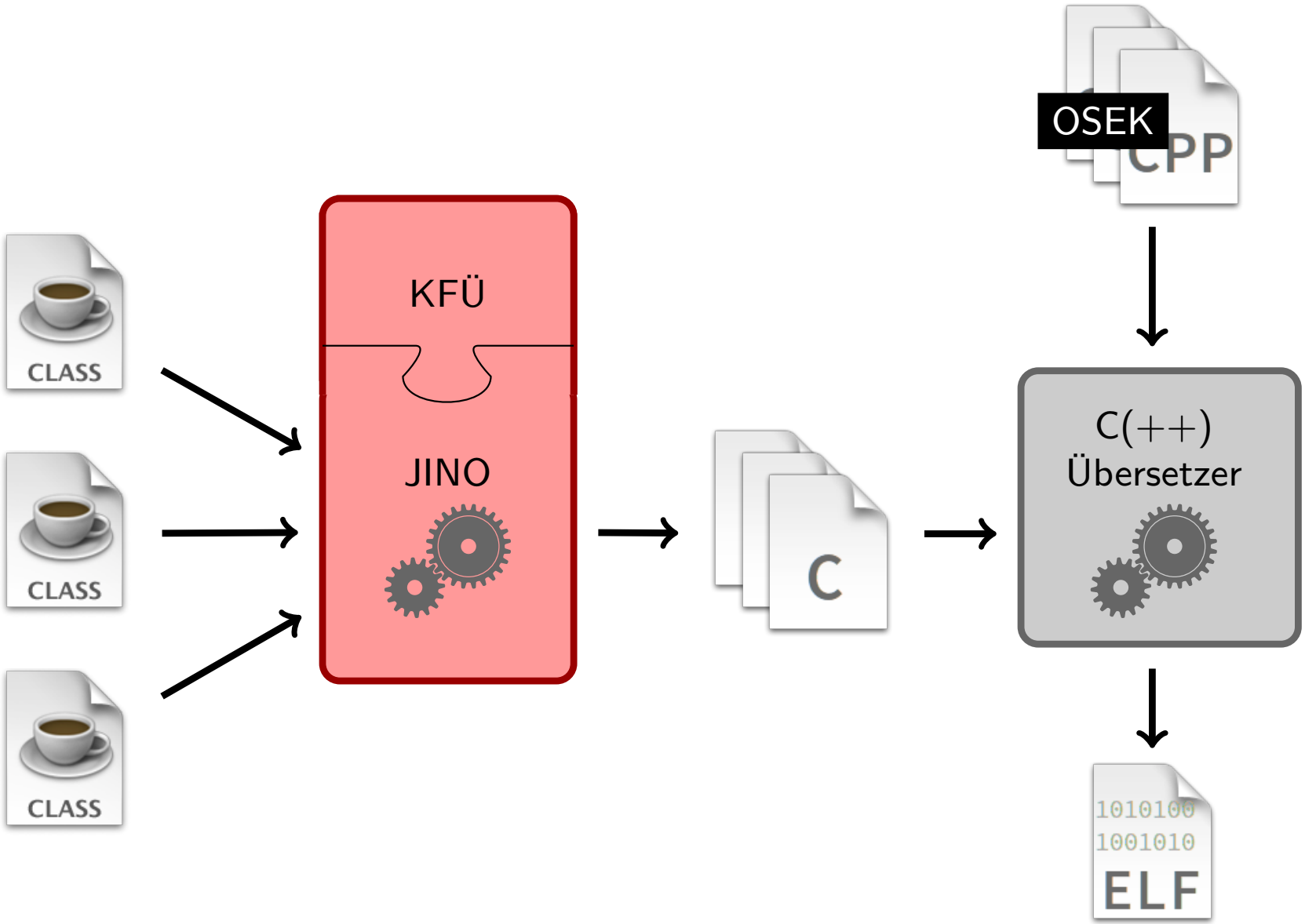
KESO-Bauprozess



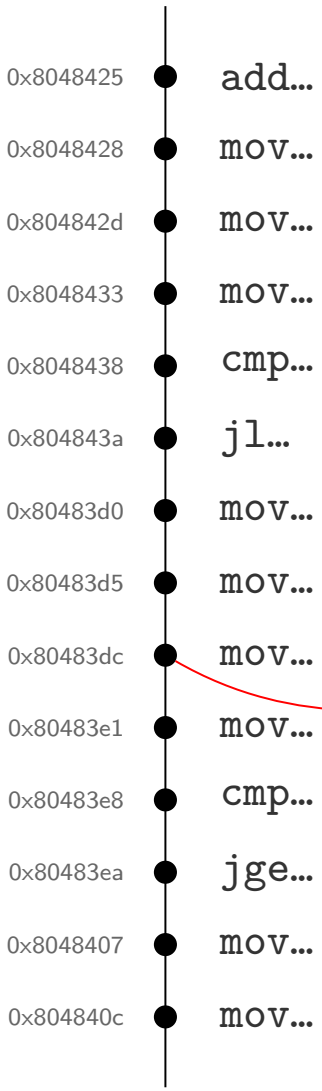
KESO-Bauprozess



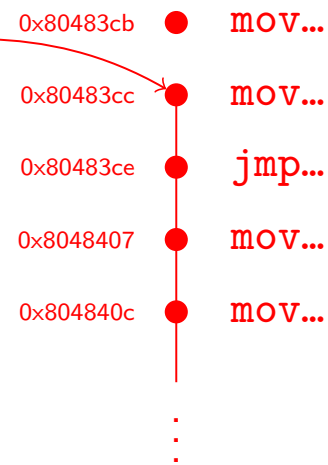
KESO-Bauprozess



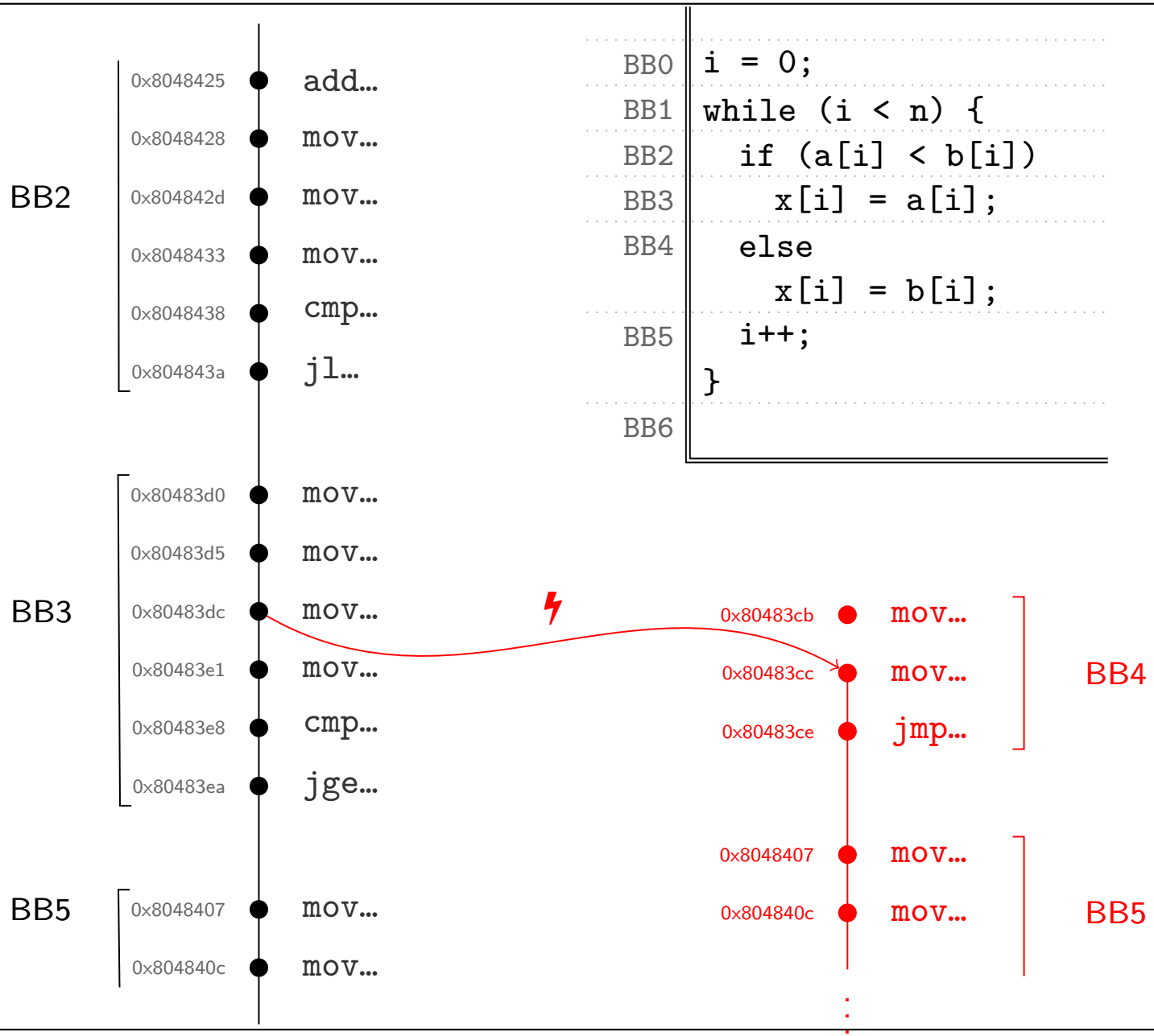
Kontrollflussüberwachung



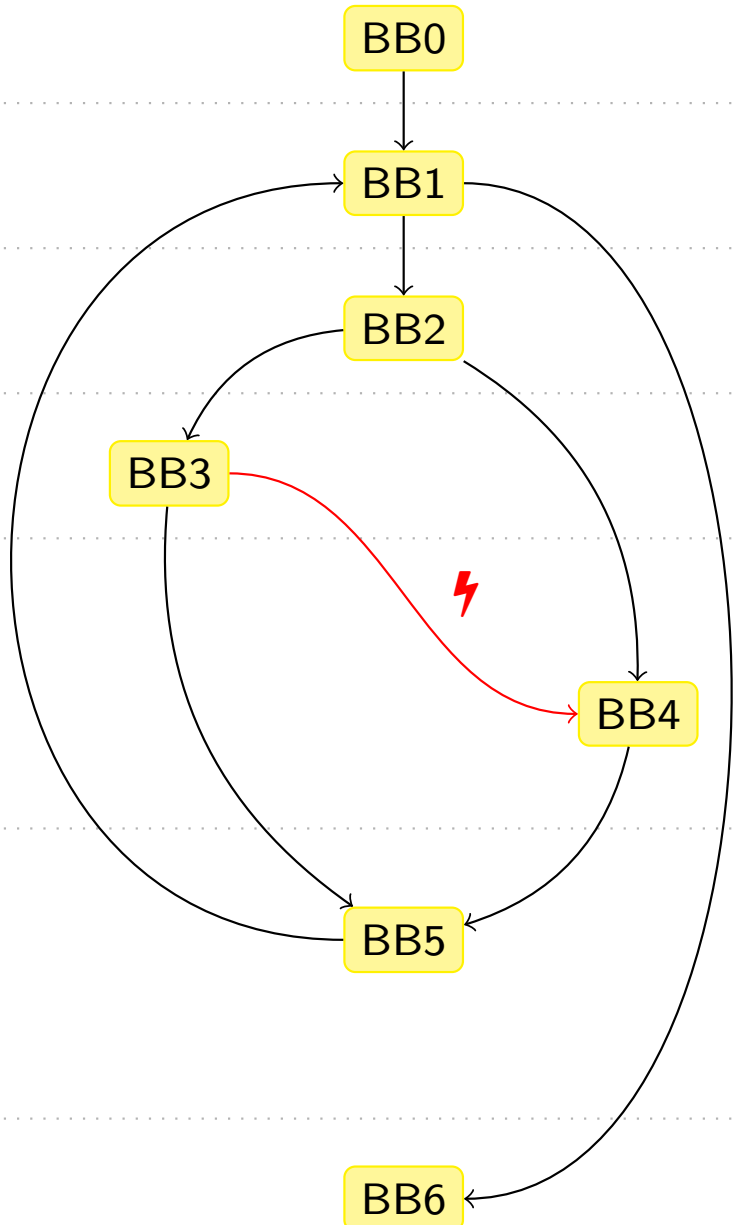

```
i = 0;
while (i < n) {
  if (a[i] < b[i])
    x[i] = a[i];
  else
    x[i] = b[i];
  i++;
}
```



Kontrollflussüberwachung



Kontrollflussüberwachung




```
BB0  i = 0;
BB1  while (i < n) {
BB2    if (a[i] < b[i])
BB3      x[i] = a[i];
BB4    else
         x[i] = b[i];
BB5    i++;
     }
BB6
```

Umgesetzte Verfahren

Basisblockebene:

- Plain Interblock Error Detection [1], [2]
- Enhanced Control-Flow Checking using Assertions (ECCA) [3]
- Control-Flow Checking by Software Signatures (CFCSS) [4]
- Yet Another Control-Flow Checking using Assertions (YACCA) [5], [6]

Dominanzregionsebene:

- Dominatorbasiertes Verfahren [7]
  - Vergabe-Strategien: „Kleinste“ und „Verteilung“
  - Schleifenoptimierung

Umgesetzte Verfahren

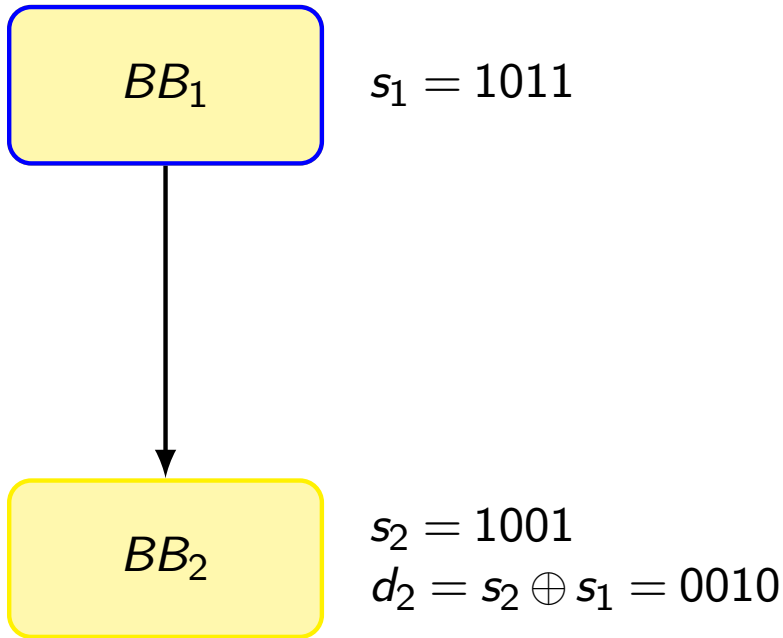
Basisblockebene:

- Plain Interblock Error Detection [1], [2]
- Enhanced Control-Flow Checking using Assertions (ECCA) [3]
- Control-Flow Checking by Software Signatures (CFCSS) [4]
- Yet Another Control-Flow Checking using Assertions (YACCA) [5], [6]

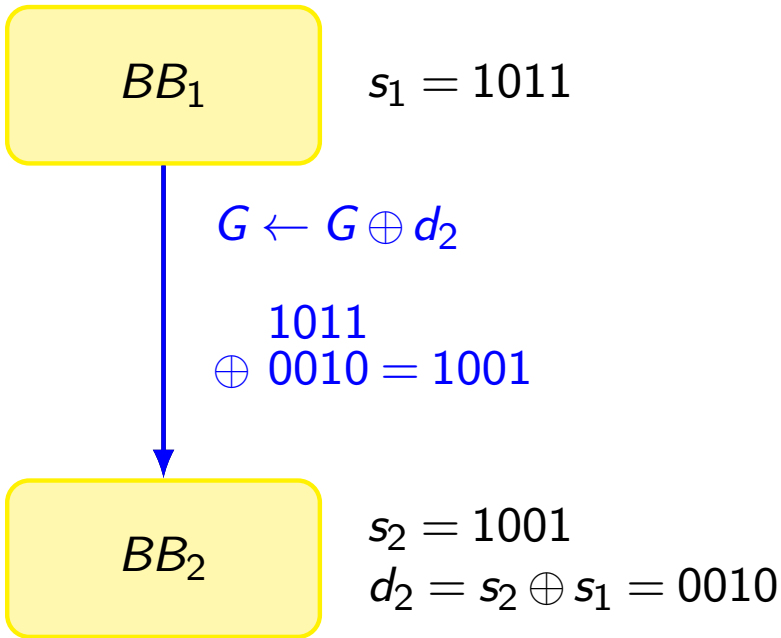
Dominanzregionsebene:

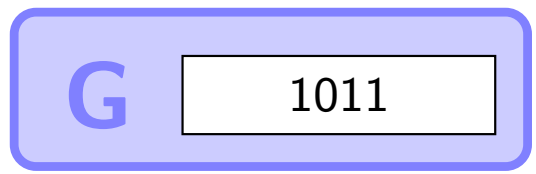
- Dominatorbasiertes Verfahren [7]
  - Vergabe-Strategien: „Kleinste“ und „Verteilung“
  - Schleifenoptimierung

Control-Flow Checking by Software Signatures

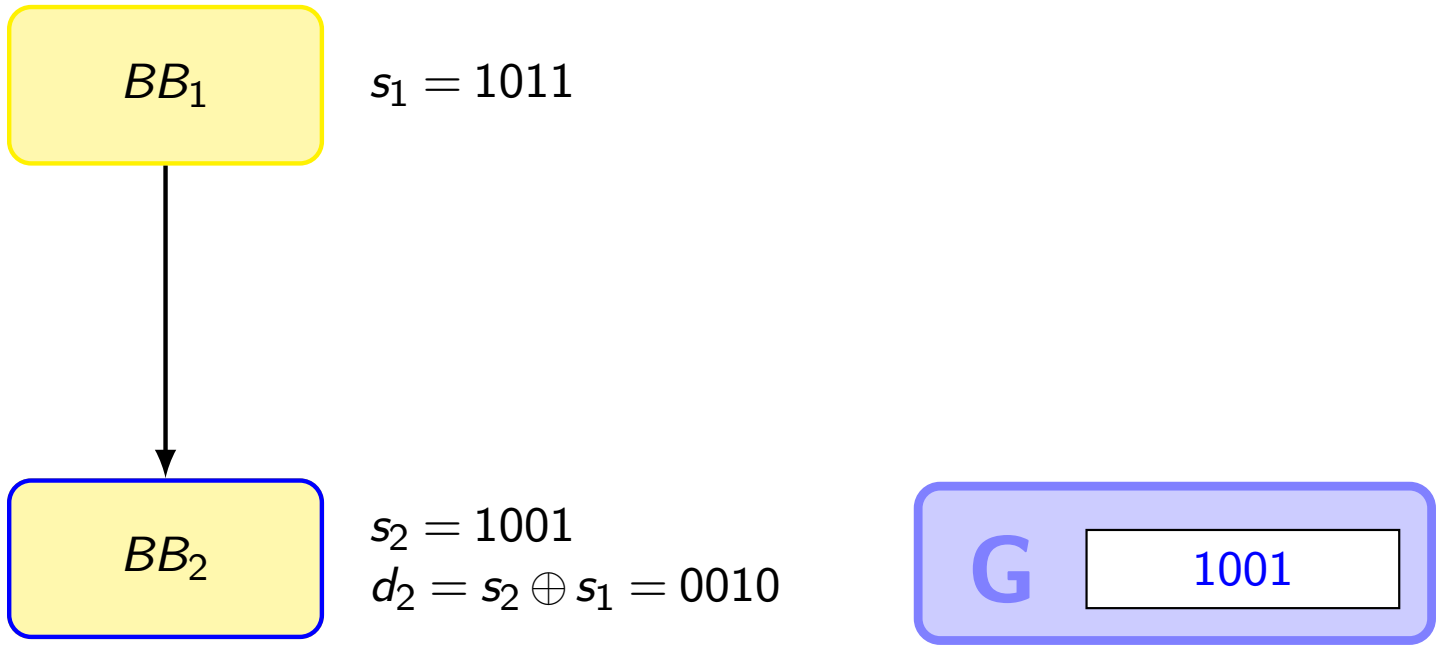


Control-Flow Checking by Software Signatures

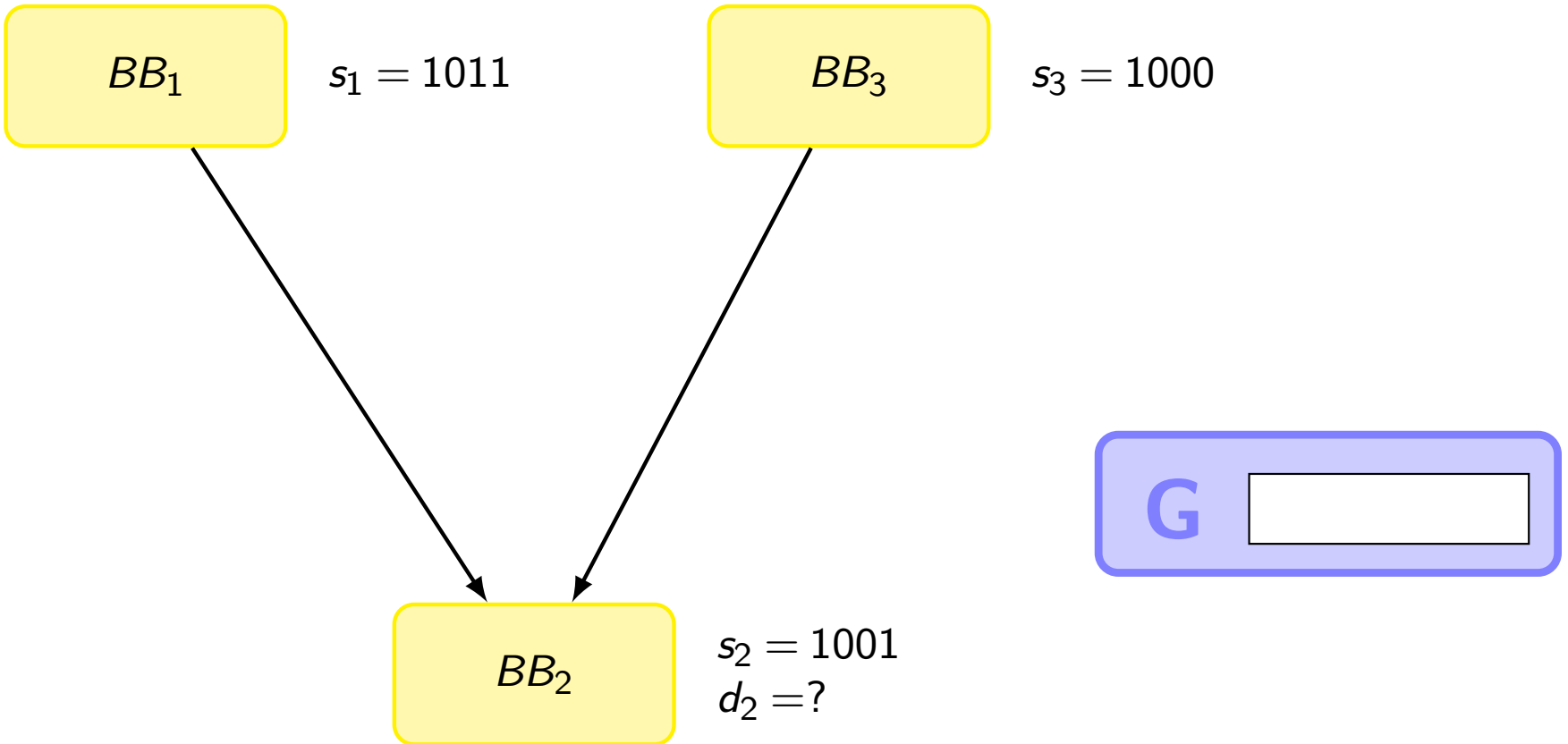




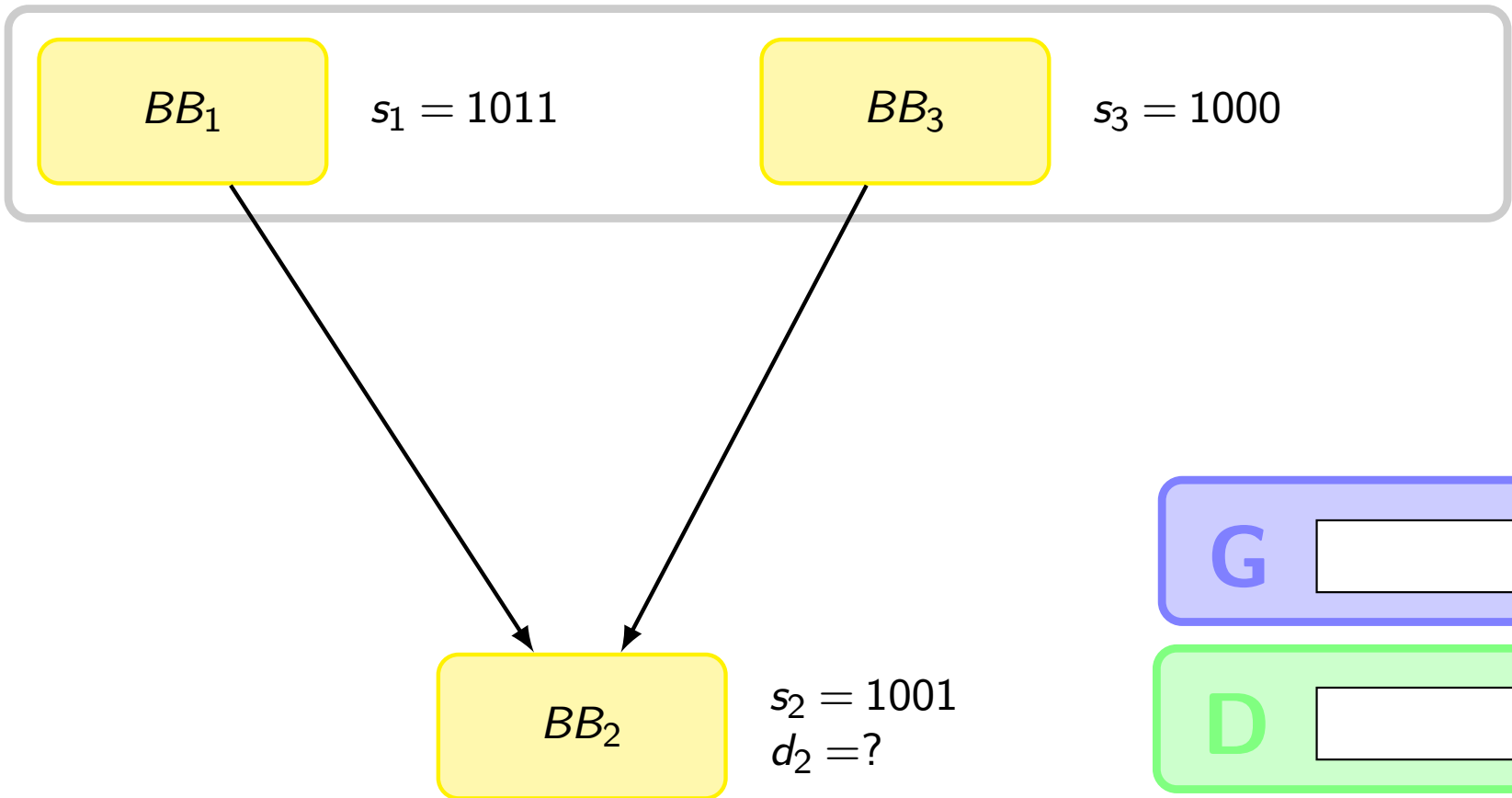
Control-Flow Checking by Software Signatures



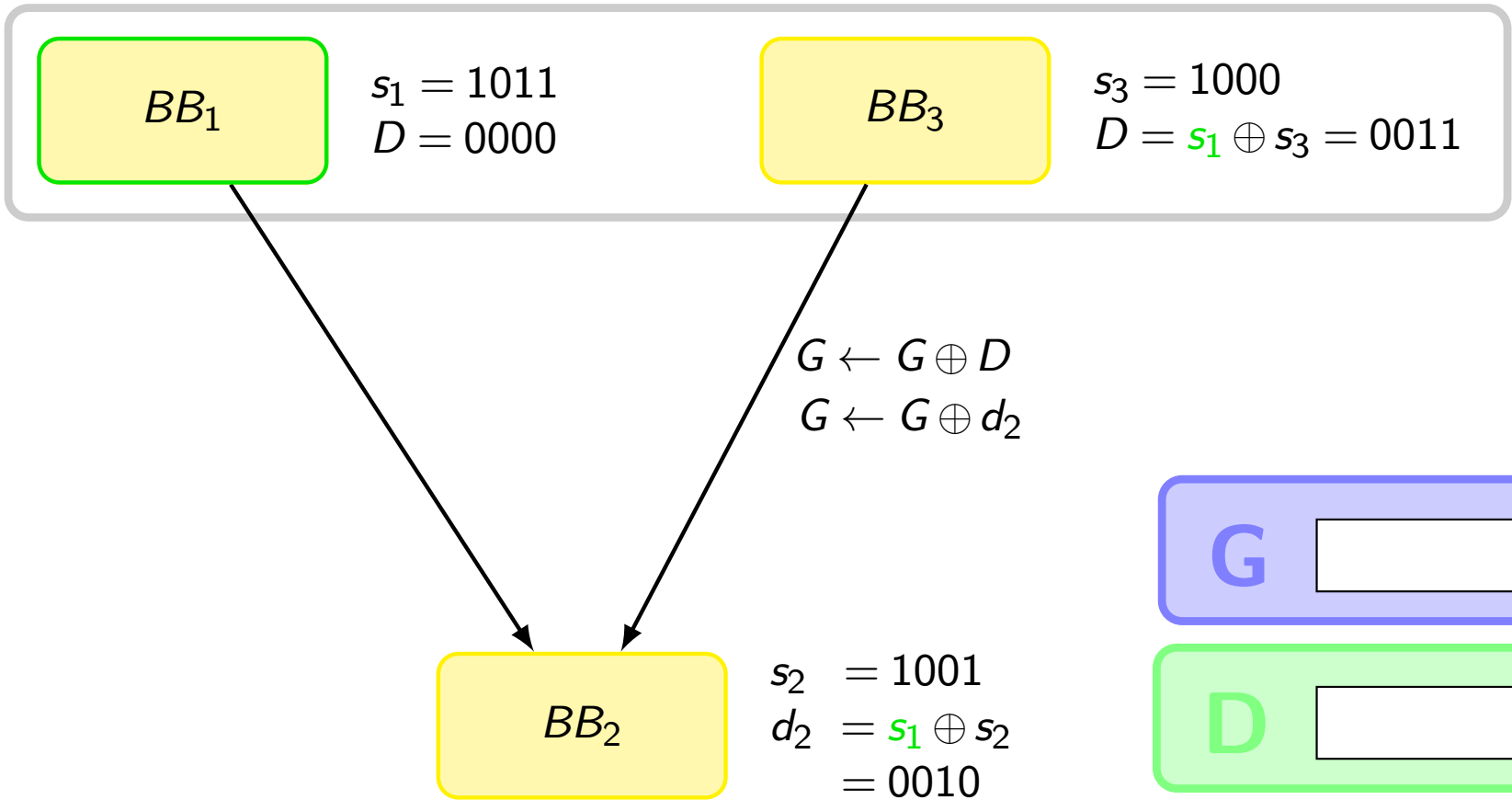
Control-Flow Checking by Software Signatures



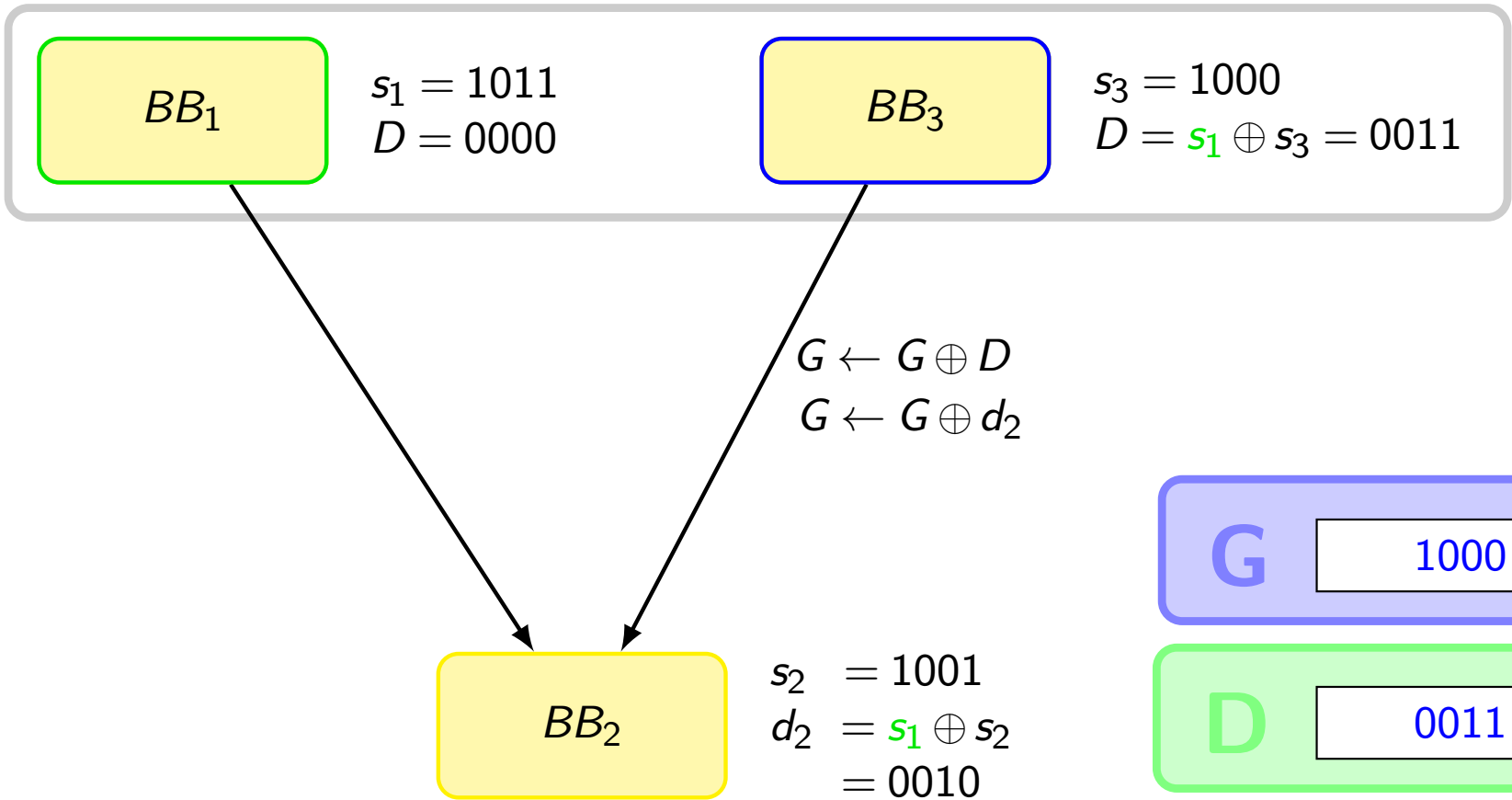
Control-Flow Checking by Software Signatures



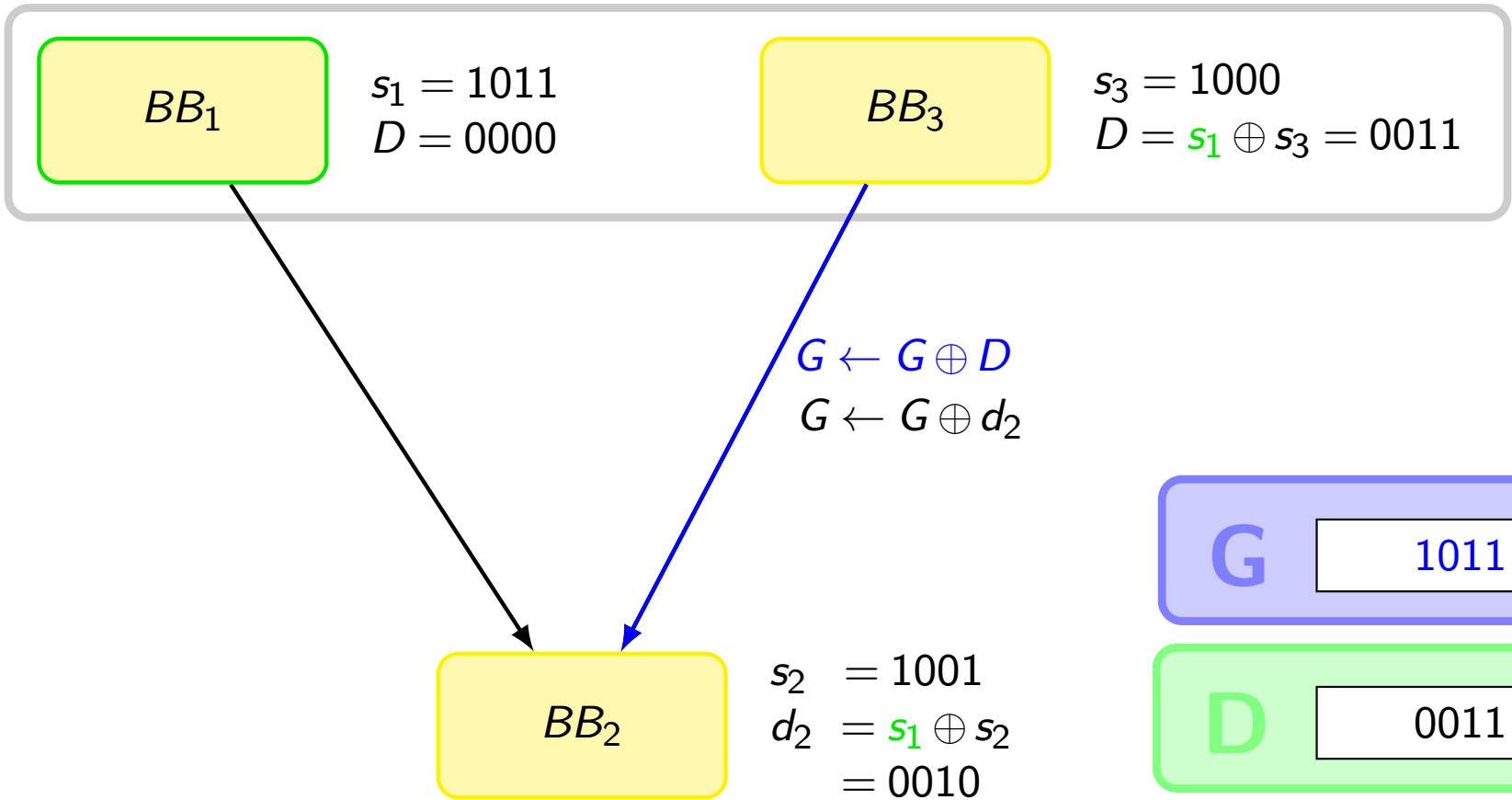
Control-Flow Checking by Software Signatures



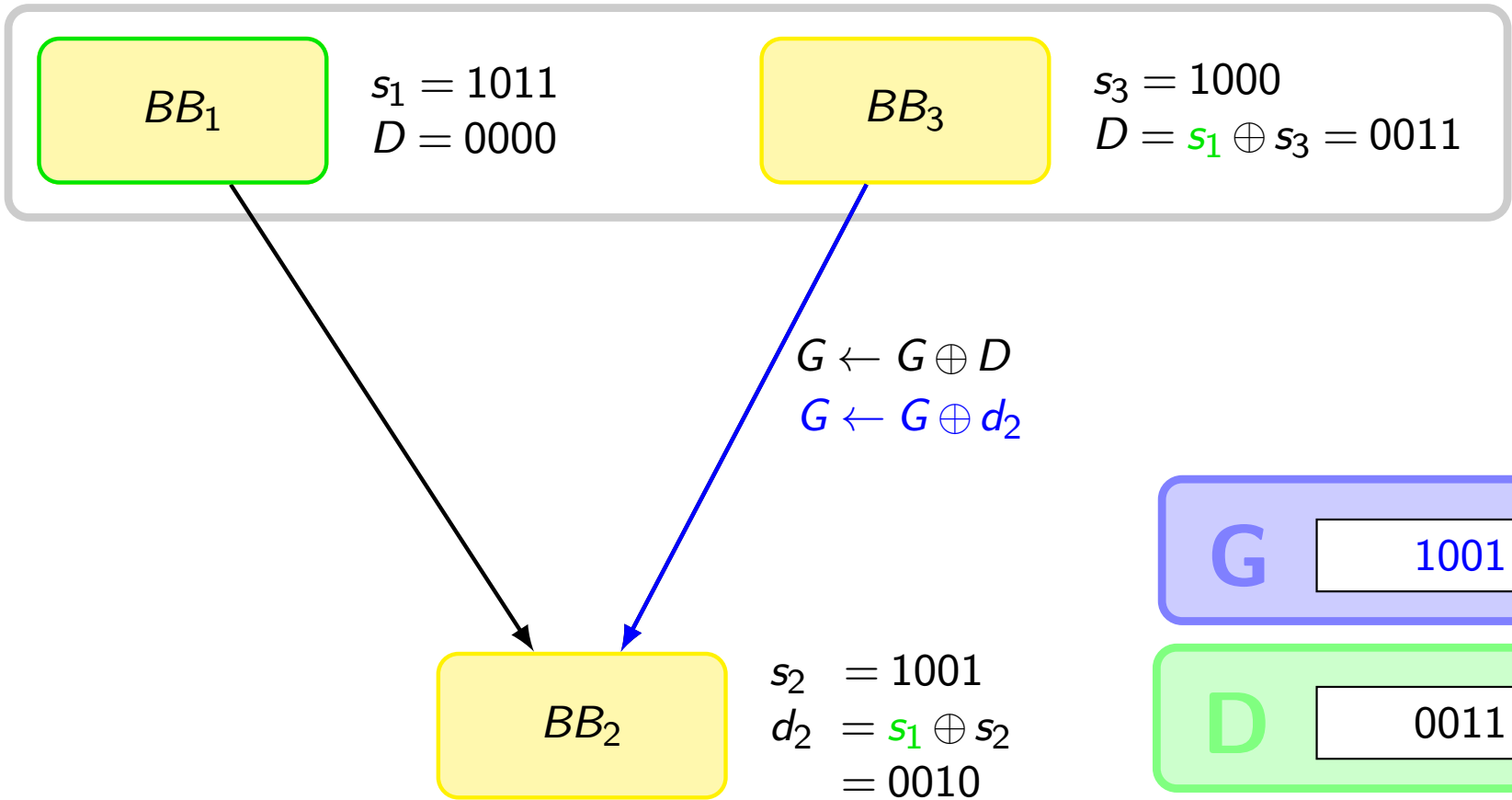
Control-Flow Checking by Software Signatures



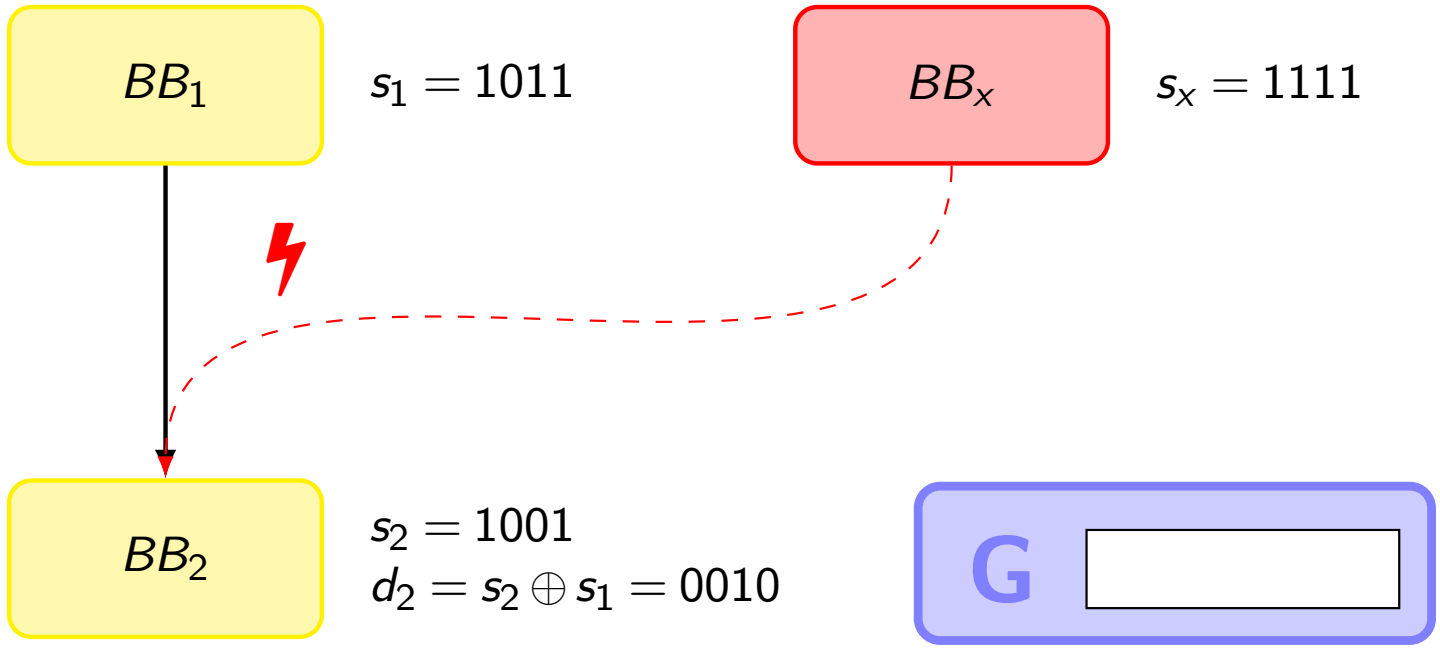
Control-Flow Checking by Software Signatures



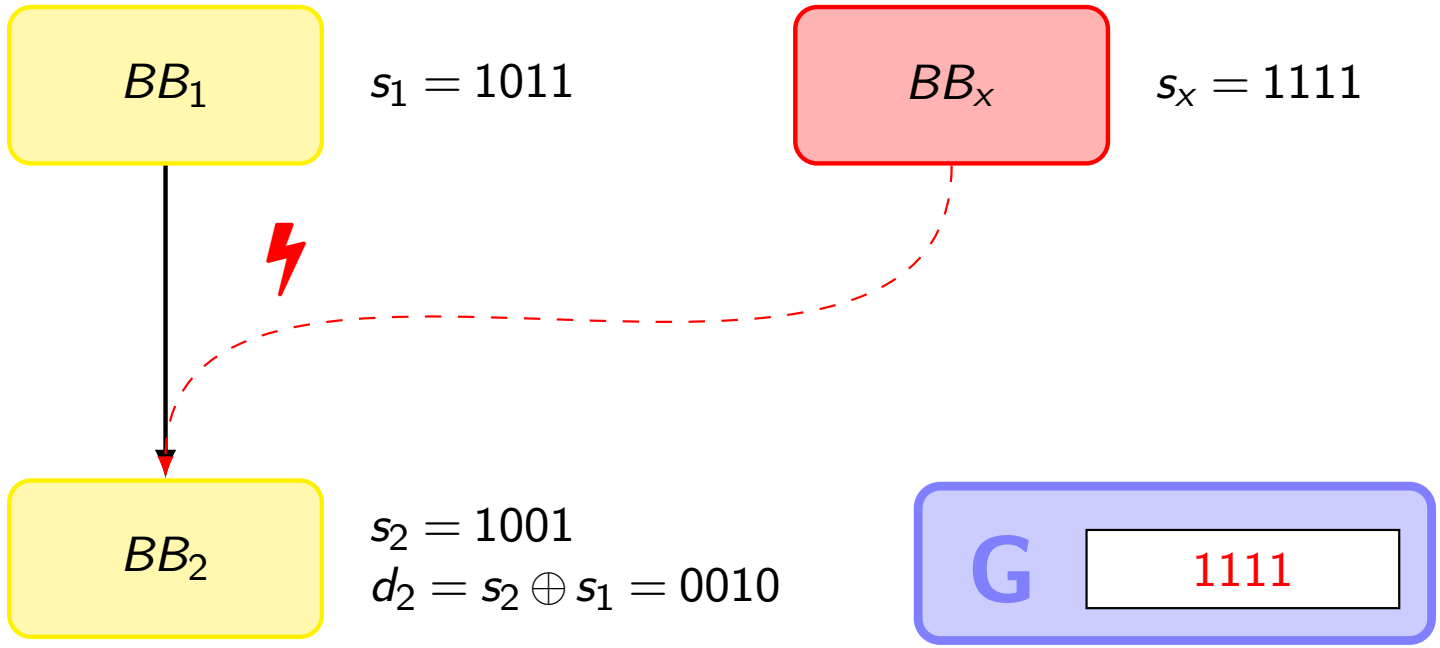
Control-Flow Checking by Software Signatures



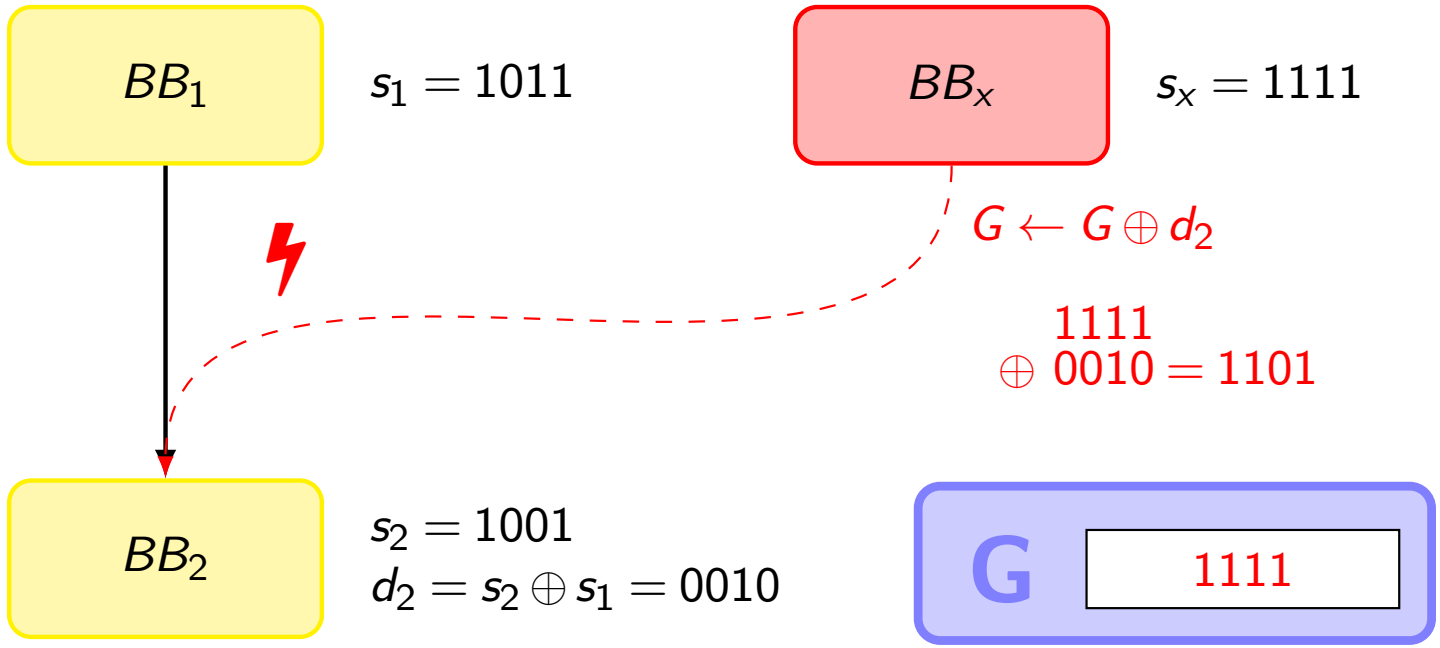
Control-Flow Checking by Software Signatures



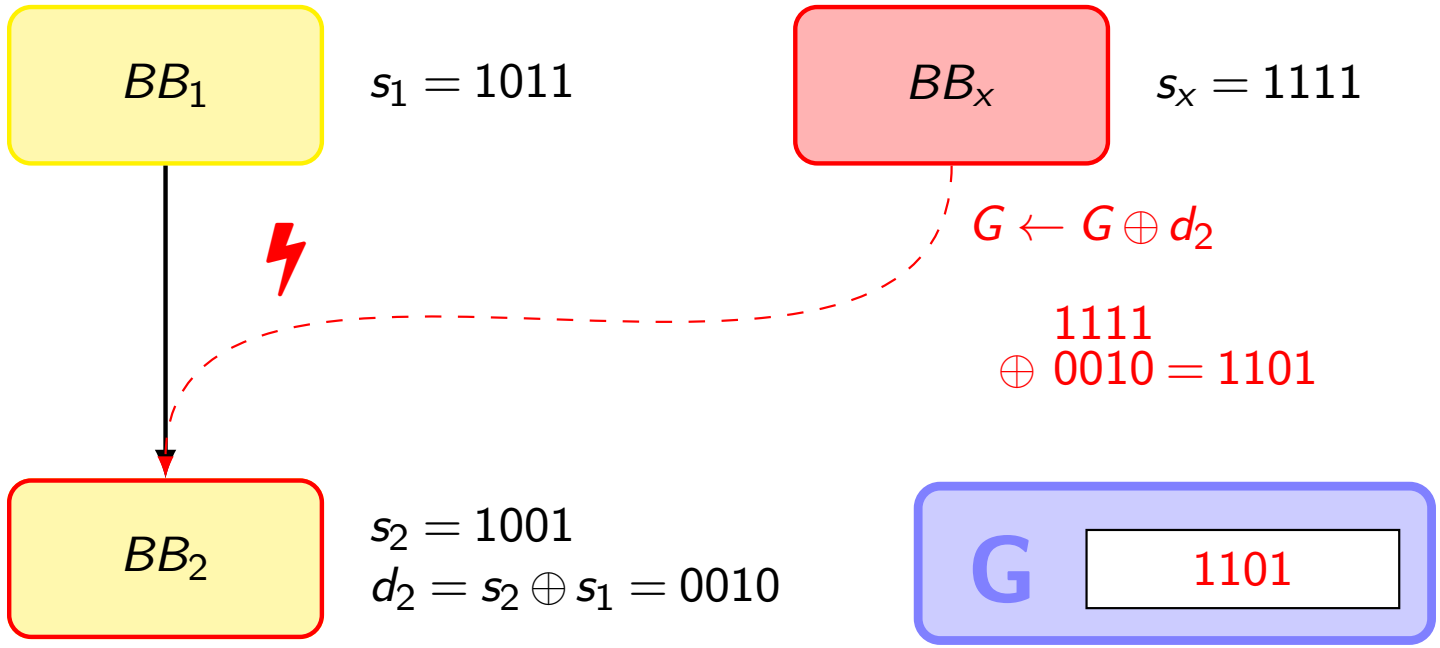
Control-Flow Checking by Software Signatures



Control-Flow Checking by Software Signatures



Control-Flow Checking by Software Signatures



Umgesetzte Verfahren

Basisblockebene:

- Plain Interblock Error Detection [1], [2]
- Enhanced Control-Flow Checking using Assertions (ECCA) [3]
- Control-Flow Checking by Software Signatures (CFCSS) [4]
- Yet Another Control-Flow Checking using Assertions (YACCA) [5], [6]

Dominanzregionsebene:

- Dominatorbasiertes Verfahren [7]
  - Vergabe-Strategien: „Kleinste“ und „Verteilung“
  - Schleifenoptimierung

Umgesetzte Verfahren

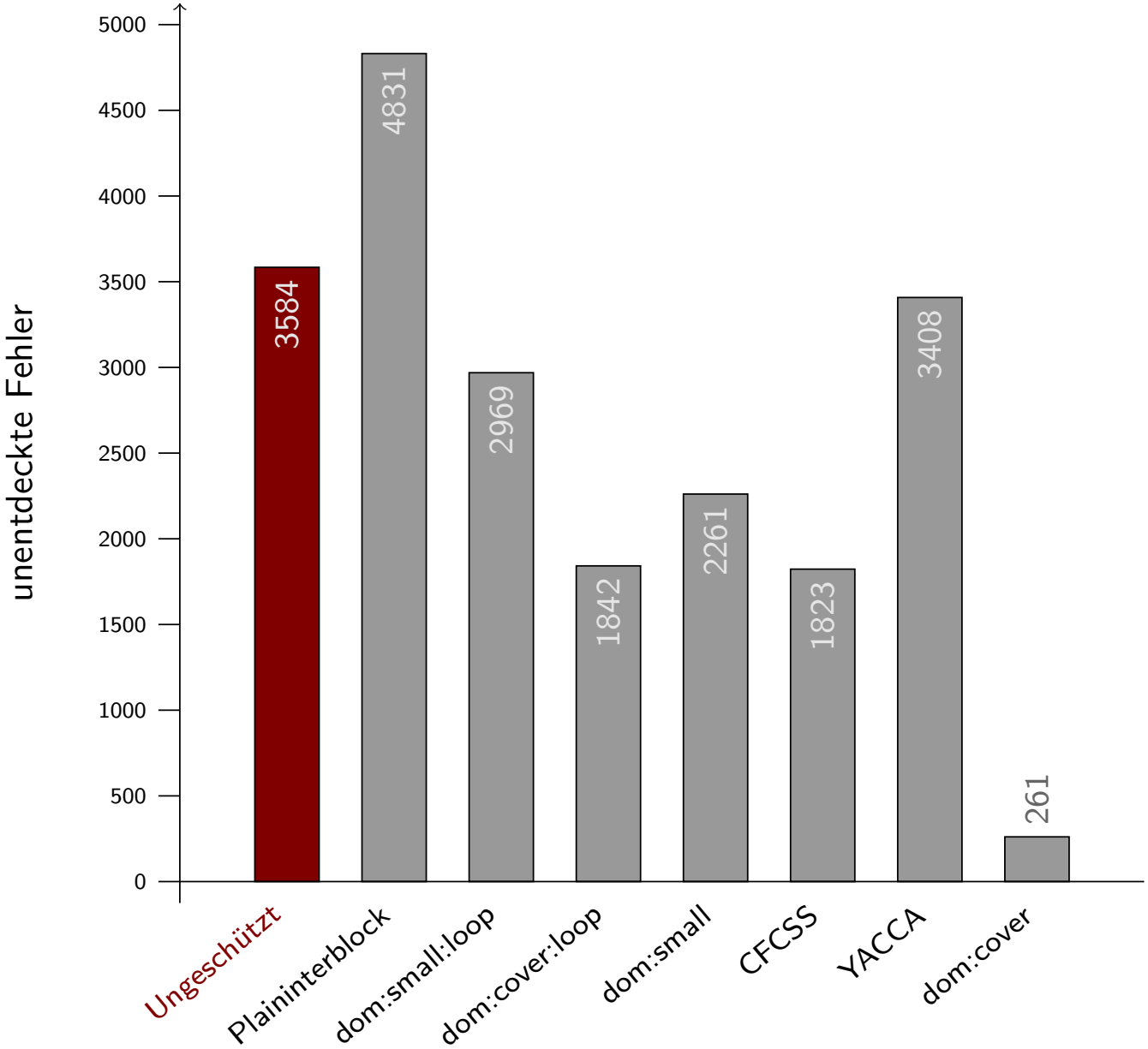
Basisblockebene:

- Plain Interblock Error Detection [1], [2]
- Enhanced Control-Flow Checking using Assertions (ECCA) [3]
- Control-Flow Checking by Software Signatures (CFCSS) [4]
- Yet Another Control-Flow Checking using Assertions (YACCA) [5], [6]

Dominanzregionsebene:

- Dominatorbasiertes Verfahren [7]
  - Vergabe-Strategien: „Kleinste“ und „Verteilung“
  - Schleifenoptimierung

Kontrollflussfehler

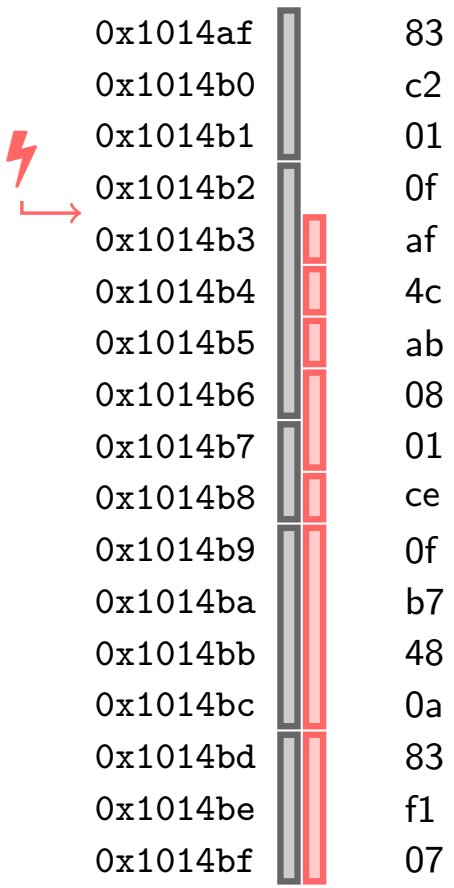


Unausgerichtete Sprünge

Ursprüngliche Instruktionsfolge

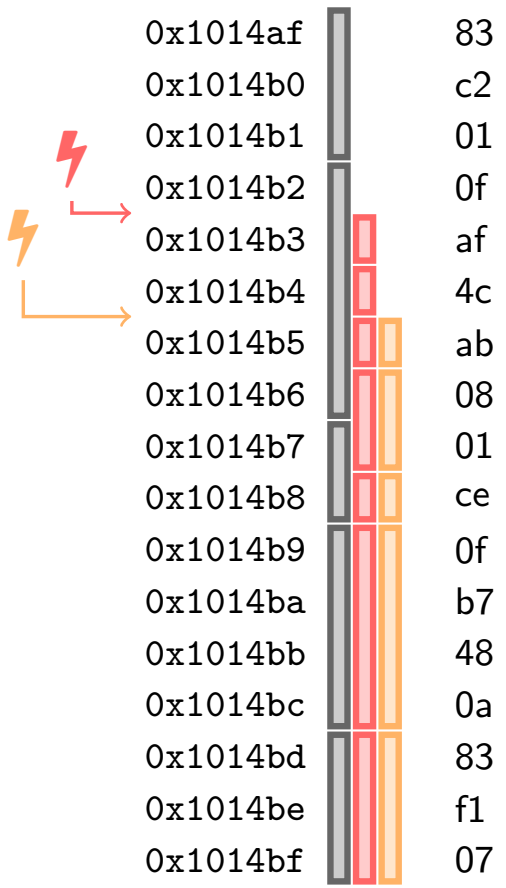
0x1014af	83	
0x1014b0	c2	`add edx,0x1`
0x1014b1	01	
0x1014b2	0f	
0x1014b3	af	`imul ecx,DWORD PTR`
0x1014b4	4c	`[ebx+ebp*4+0x8]`
0x1014b5	ab	
0x1014b6	08	
0x1014b7	01	`add esi,ecx`
0x1014b8	ce	
0x1014b9	0f	
0x1014ba	b7	`movzx ecx,WORD PTR`
0x1014bb	48	`[eax+0xa]`
0x1014bc	0a	
0x1014bd	83	
0x1014be	f1	`xor ecx,0x7`
0x1014bf	07	

Unausgerichtete Sprünge



		Ursprüngliche Instruktionsfolge	Unausgerichtete Instruktionsfolge
0x1014af	83		
0x1014b0	c2	`add edx,0x1`	
0x1014b1	01		
0x1014b2	0f		
0x1014b3	af	`imul ecx,DWORD PTR`	`scas eax,DWORD PTR es:[edi]`
0x1014b4	4c	`[ebx+ebp*4+0x8]`	`dec esp`
0x1014b5	ab		`stos DWORD PTR es:[edi],eax`
0x1014b6	08		`or BYTE PTR [ecx],al`
0x1014b7	01	`add esi,ecx`	
0x1014b8	ce		`into`
0x1014b9	0f		
0x1014ba	b7	`movzx ecx,WORD PTR`	`movzx ecx,WORD PTR`
0x1014bb	48	`[eax+0xa]`	`[eax+0xa]`
0x1014bc	0a		
0x1014bd	83		
0x1014be	f1	`xor ecx,0x7`	`xor ecx,0x7`
0x1014bf	07		

Unausgerichtete Sprünge

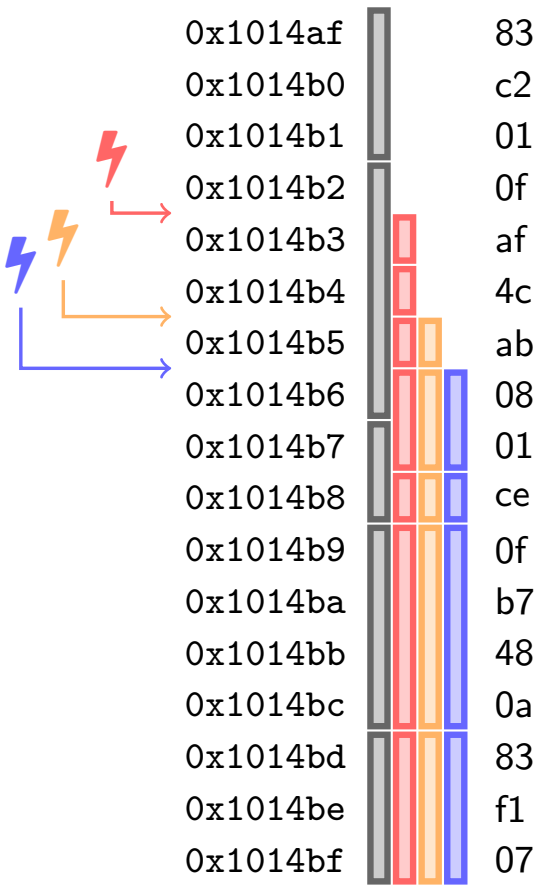


Ursprüngliche Instruktionsfolge	Unausgerichtete Instruktionsfolge
`add edx,0x1`	

`imul ecx,DWORD PTR [ebx+ebp*4+0x8]`	`scas eax,DWORD PTR es:[edi]`
	`dec esp`
	`stos DWORD PTR es:[edi],eax`

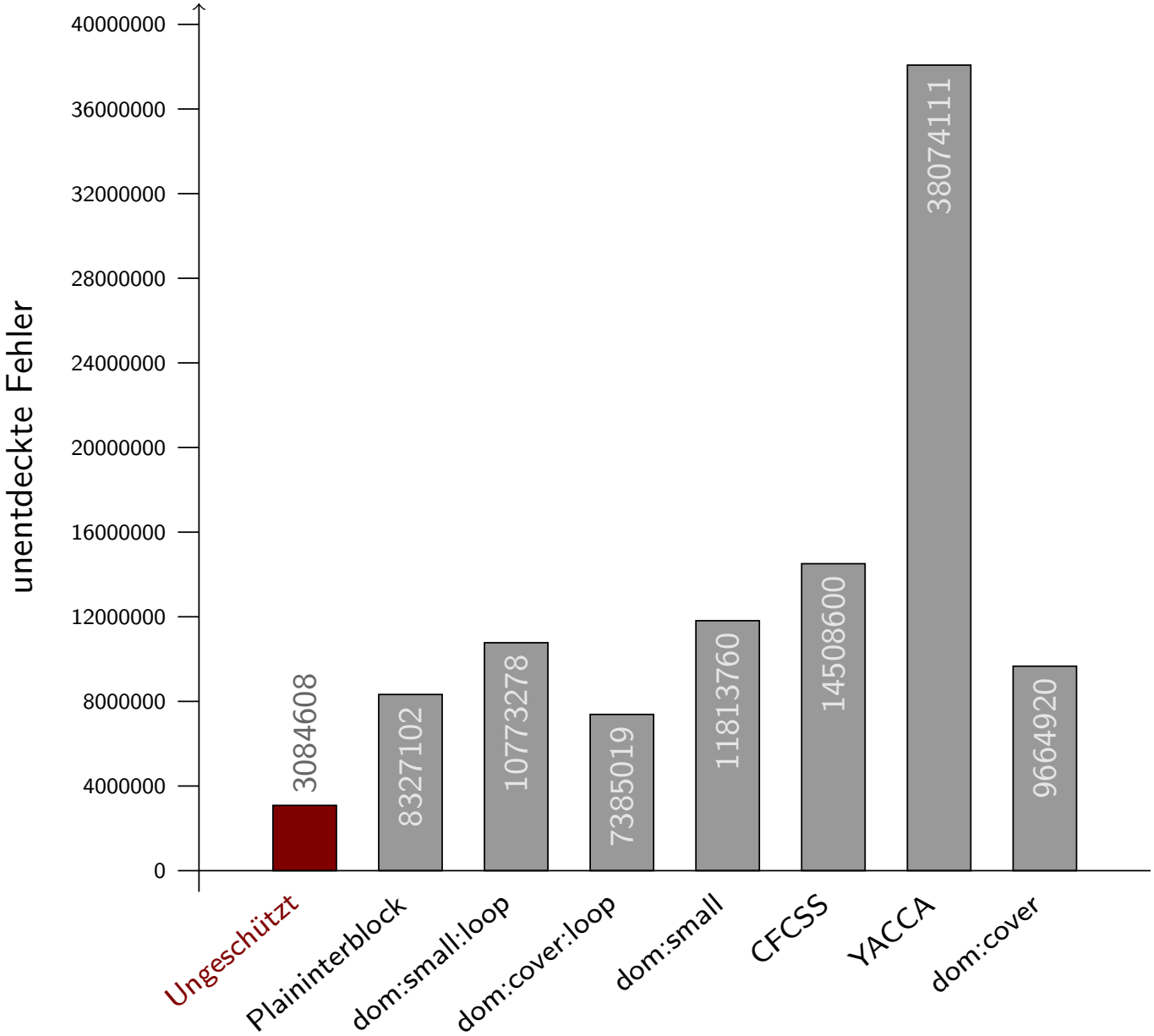
`add esi,ecx`	`or BYTE PTR [ecx],al` `into`
`movzx ecx,WORD PTR [eax+0xa]`	`movzx ecx,WORD PTR [eax+0xa]`
`xor ecx,0x7`	`xor ecx,0x7`

Unausgerichtete Sprünge

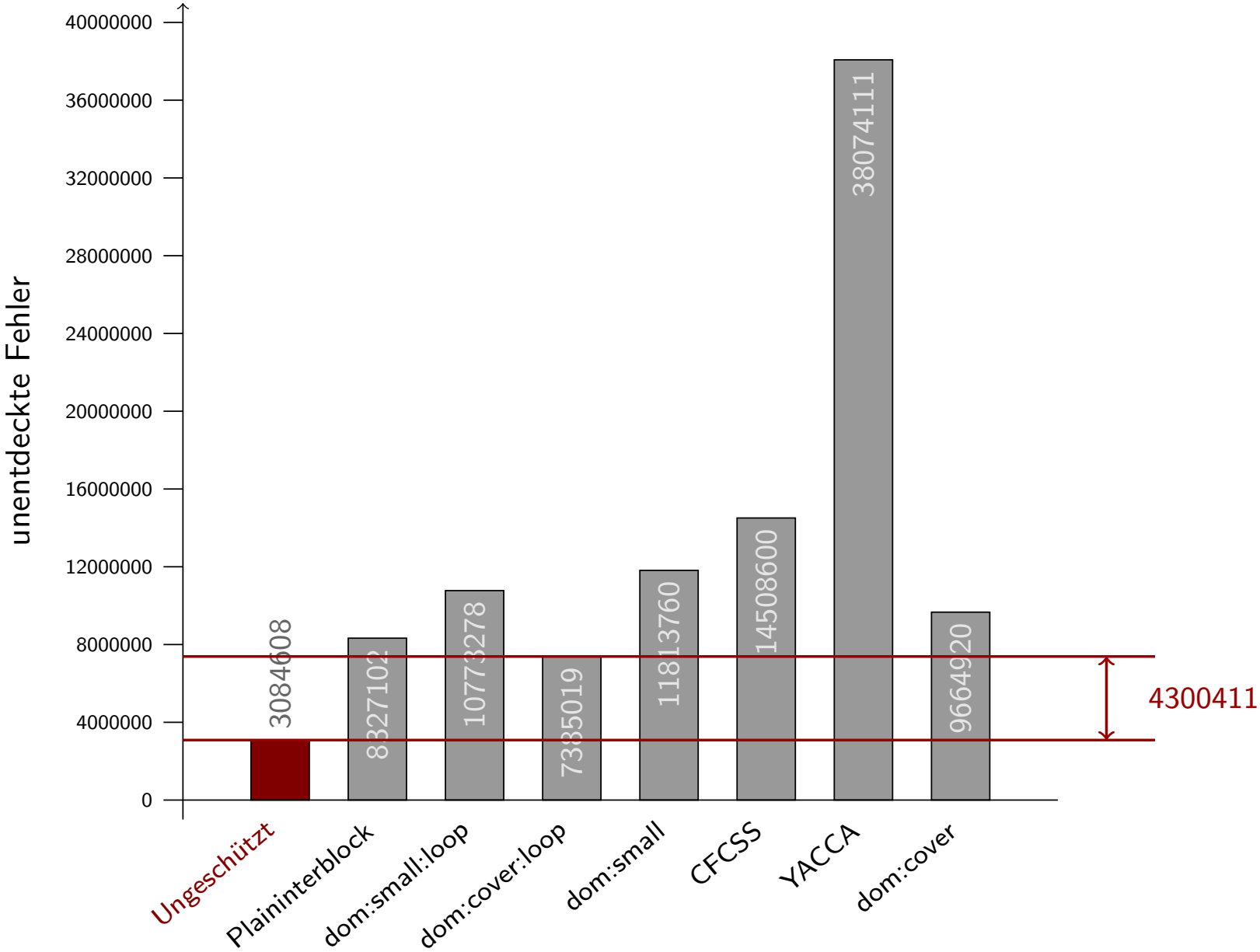


Ursprüngliche Instruktionsfolge	Unausgerichtete Instruktionsfolge
`add edx,0x1`	
`imul ecx,DWORD PTR [ebx+ebp*4+0x8]`	`scas eax,DWORD PTR es:[edi]`
	`dec esp`
	`stos DWORD PTR es:[edi],eax`
	`or BYTE PTR [ecx],al`
`add esi,ecx`	
	`into`
`movzx ecx,WORD PTR [eax+0xa]`	`movzx ecx,WORD PTR [eax+0xa]`
`xor ecx,0x7`	`xor ecx,0x7`

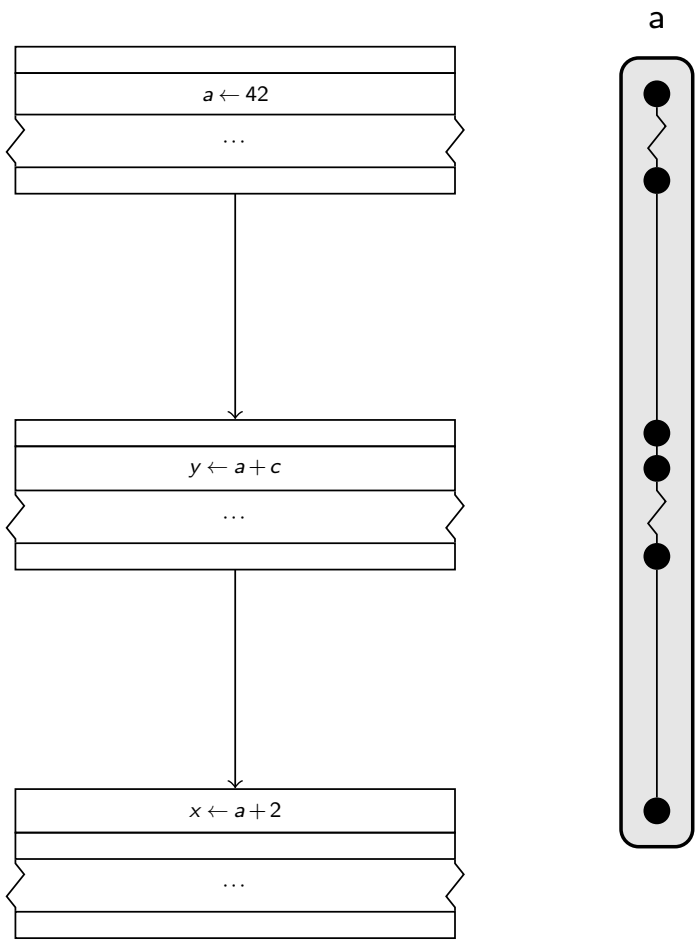
Allgemeine Einbitfehler



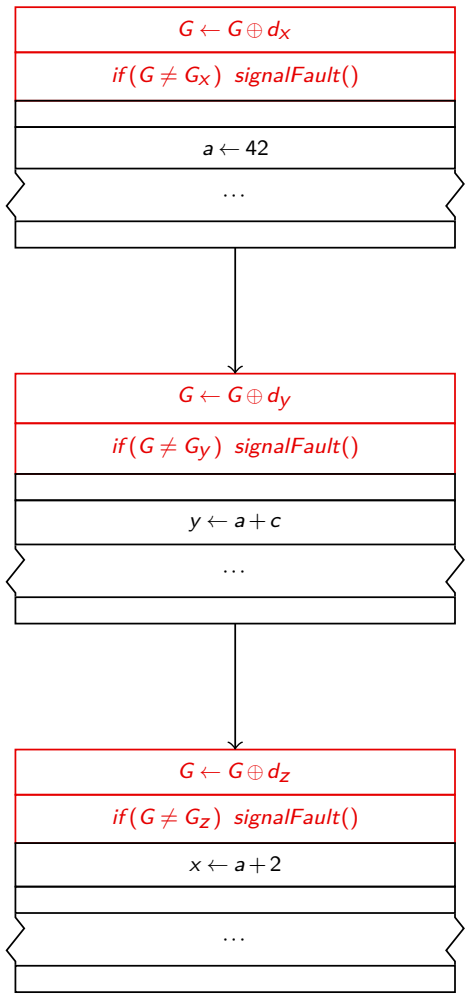
Allgemeine Einbitfehler

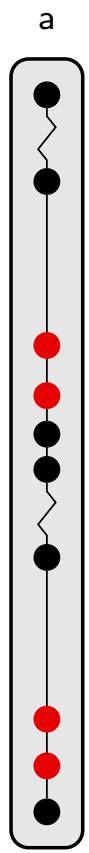


Datenfehler

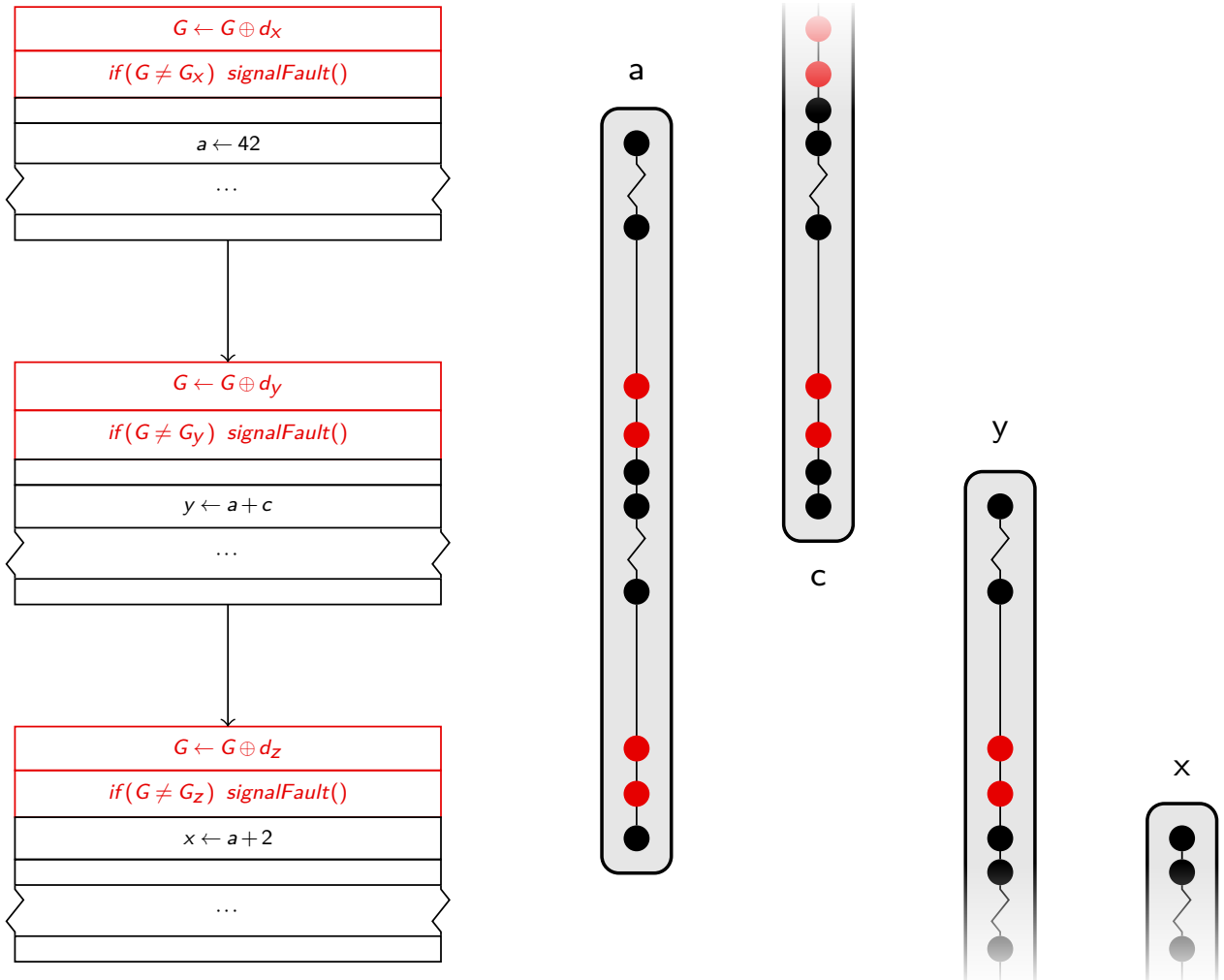


Datenfehler

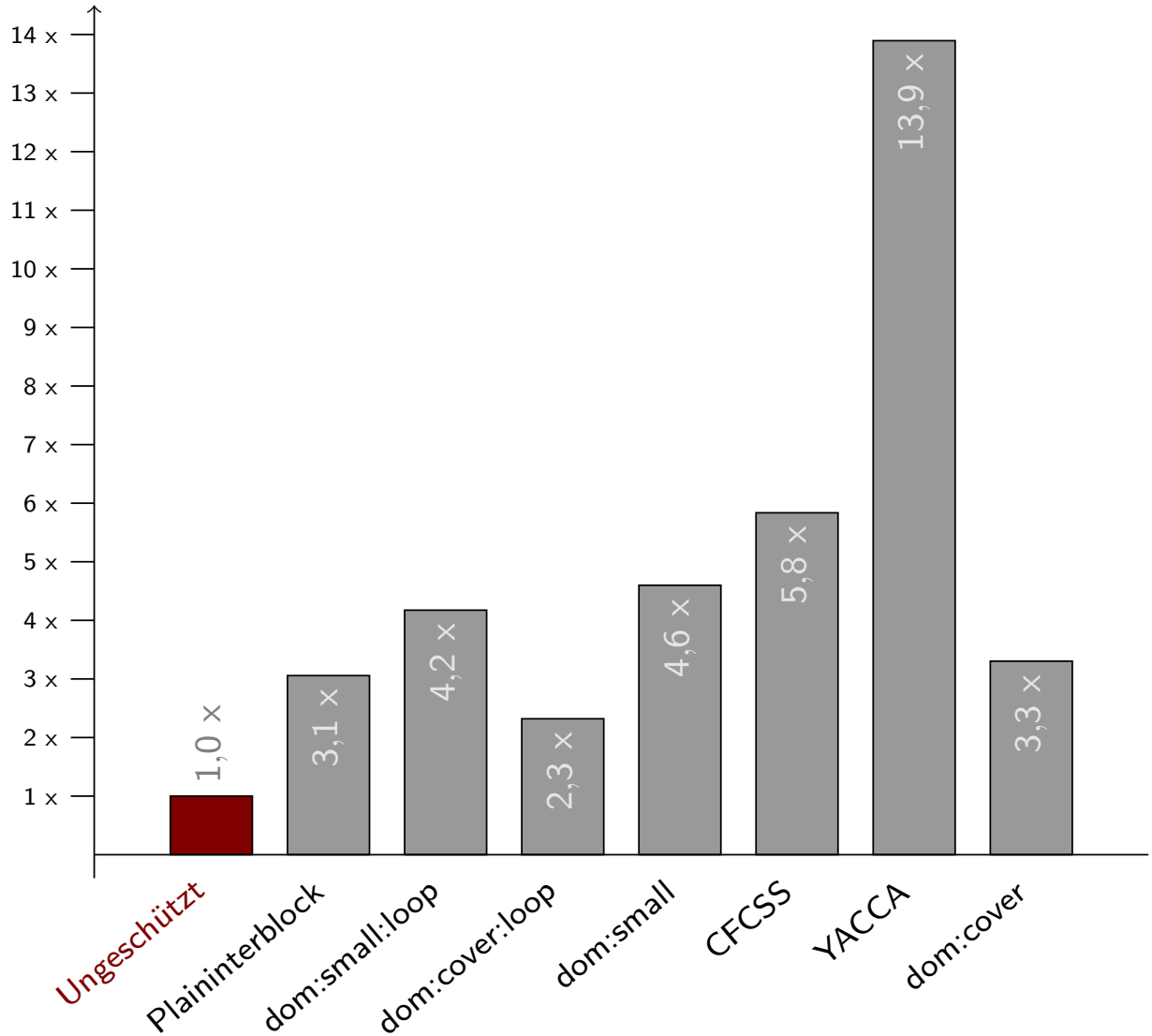




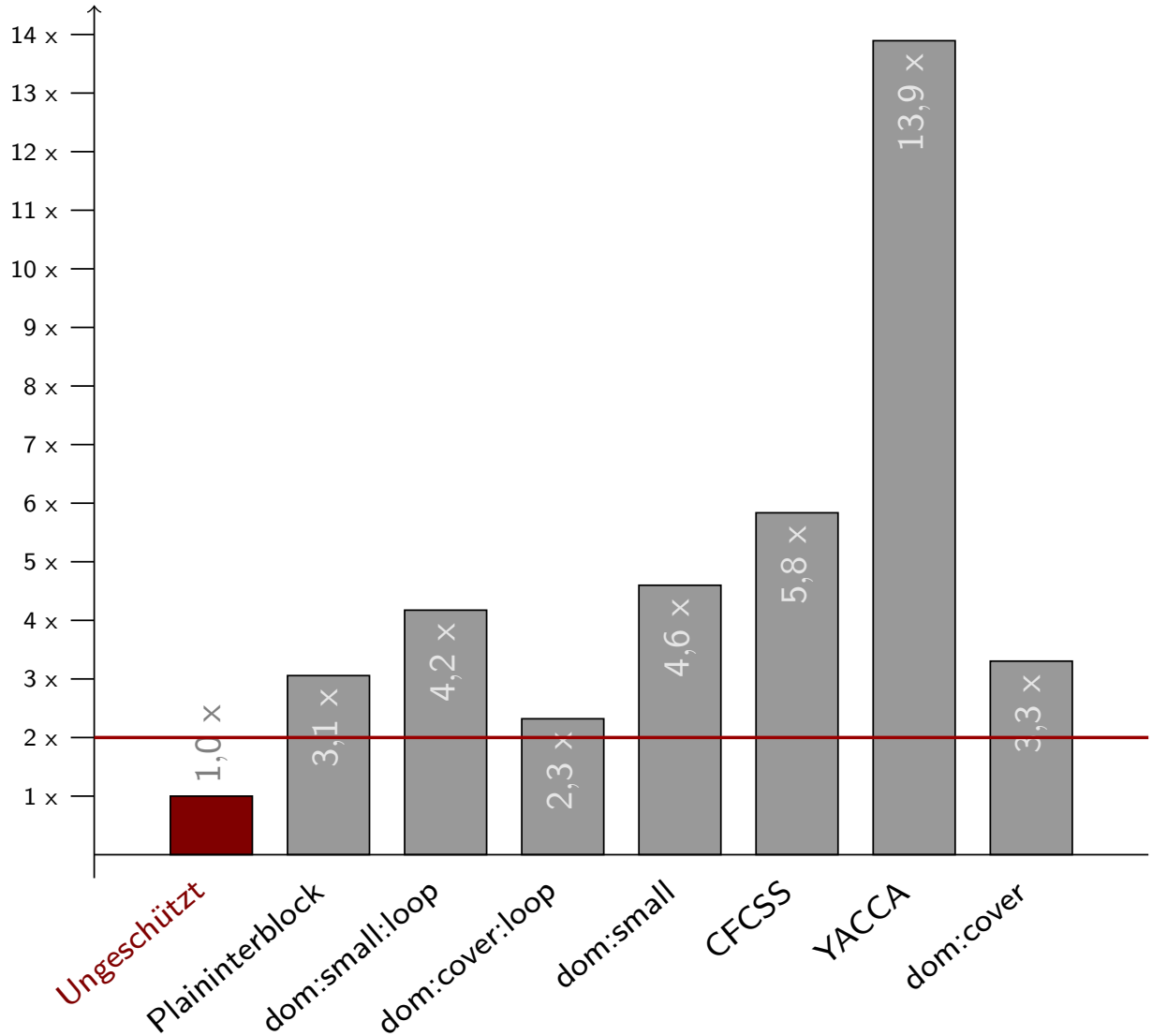
Datenfehler



Laufzeit


14 x
13 x
12 x
11 x
10 x
9 x
8 x
7 x
6 x
5 x
4 x
3 x
2 x
1 x
1,0 x
3,1 x
4,2 x
2,3 x
4,6 x
5,8 x
13,9 x
3,3 x
Ungeschützt
Plaininterblock
dom:small:loop
dom:cover:loop
dom:small
CFCSS
YACCA
dom:cover

Laufzeit

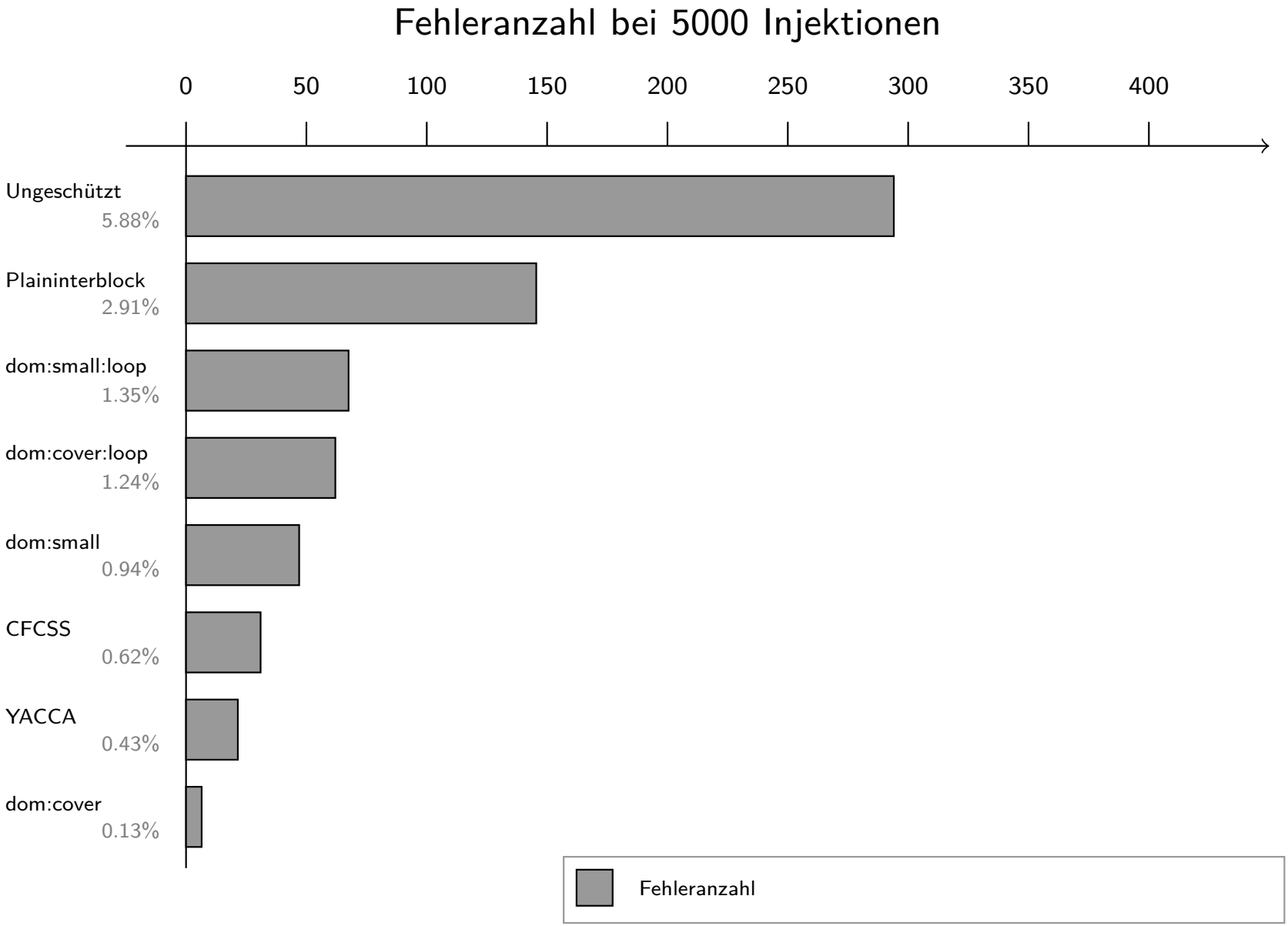

14 x
13 x
12 x
11 x
10 x
9 x
8 x
7 x
6 x
5 x
4 x
3 x
2 x
1 x
1,0 x
3,1 x
4,2 x
2,3 x
4,6 x
5,8 x
13,9 x
3,3 x
Ungeschützt
Plaininterblock
dom:small:loop
dom:cover:loop
dom:small
CFCSS
YACCA
dom:cover

Literaturvergleich

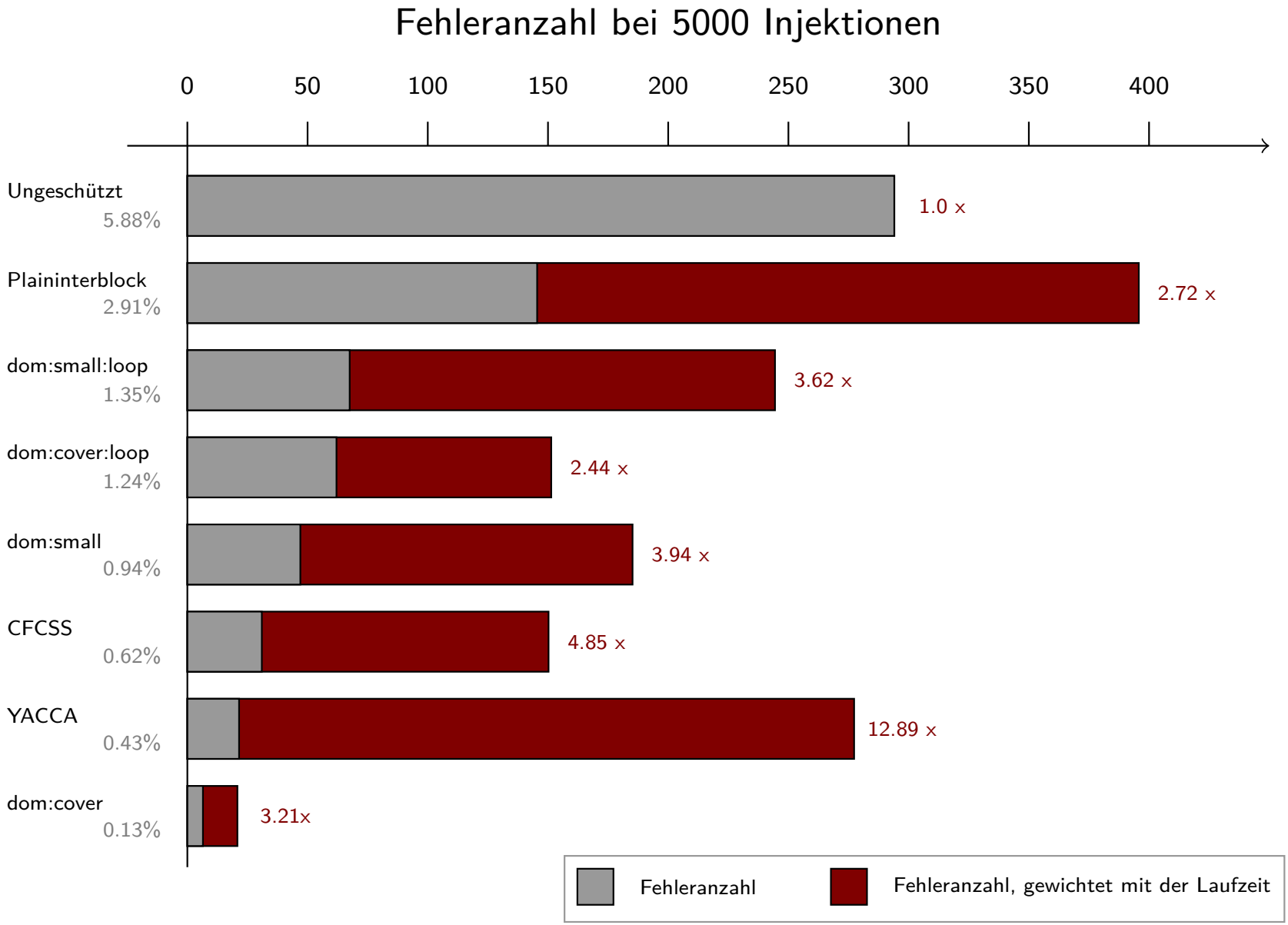
- IA32 / TC1796
- g++, v4.9.1, v4.6.3
- Optionen: -O2
- Signatur als `volatile` Variable
- Speicherzugriff TC1796: 2 Zyklen
- ...

- R4400 (MIPS), Intel 8051, TIMA T225, ...
- ?
- ?
- ?
- ?
- ...

Fehlerrate und -abdeckung



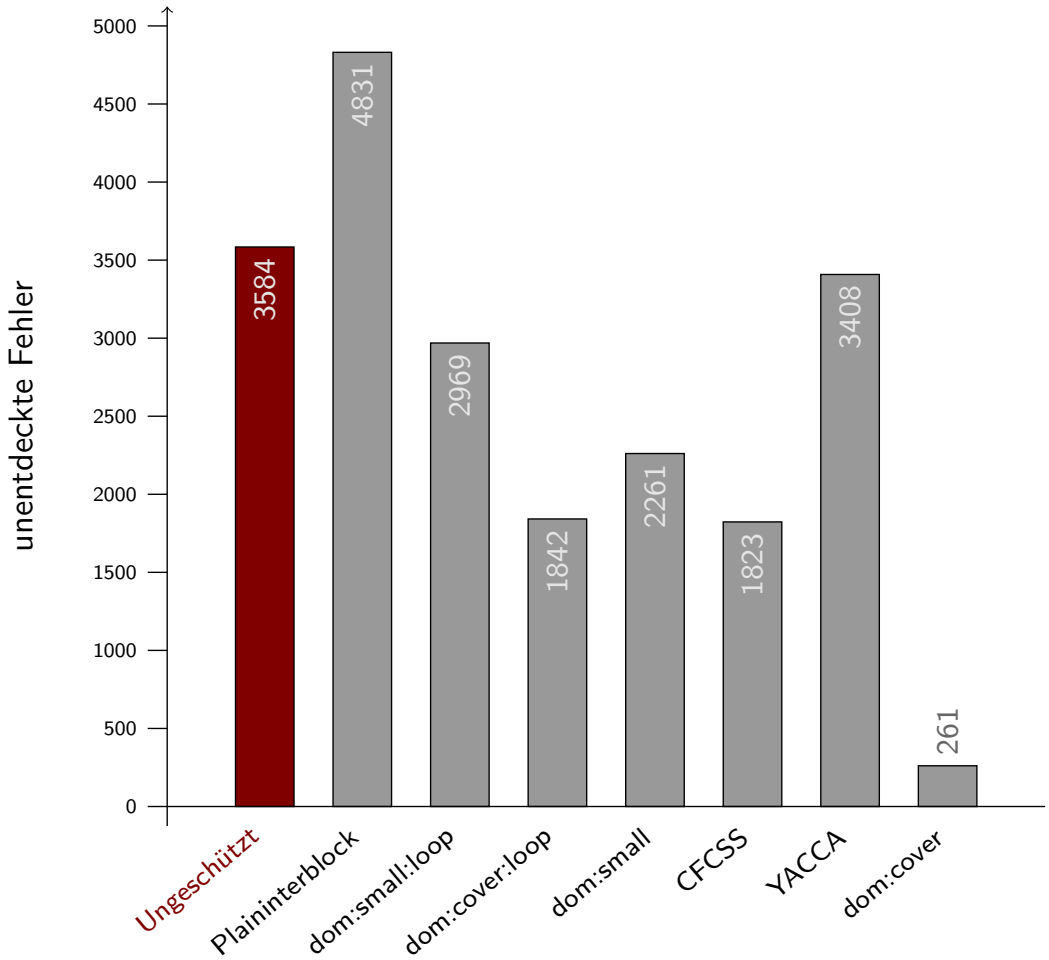
Fehlerrate und -abdeckung



Fazit

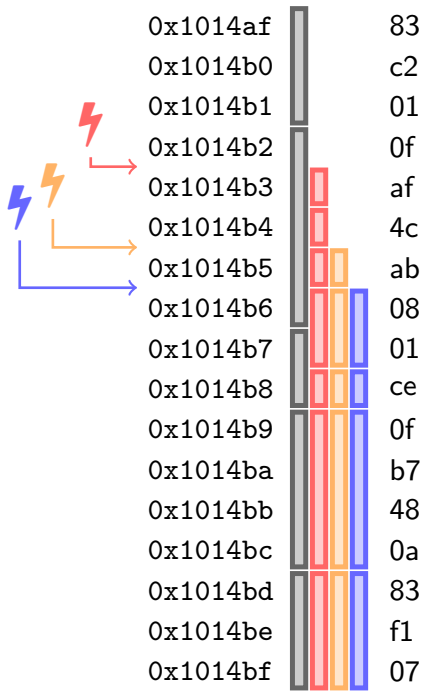
Fazit

Kontrollflussfehler



Kontrollflussfehler

Unausgerichtete Sprünge



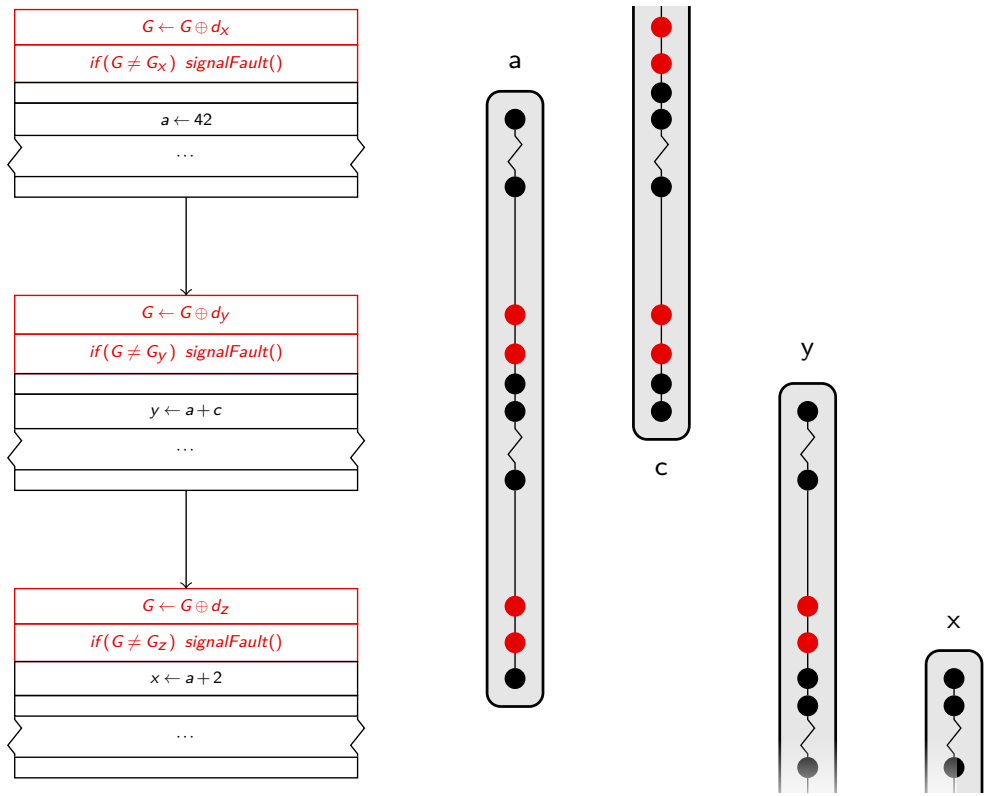
		Ursprüngliche Instruktionsfolge	Unausgerichtete Instruktionsfolge
0x1014af	83		
0x1014b0	c2	add edx,0x1	
0x1014b1	01		
0x1014b2	0f		
0x1014b3	af	imul ecx,DWORD PTR	scas eax,DWORD PTR es:[edi]
0x1014b4	4c	[ebx+ebp*4+0x8]	dec esp
0x1014b5	ab		stos DWORD PTR es:[edi],eax
0x1014b6	08		or BYTE PTR [ecx],al
0x1014b7	01	add esi,ecx	
0x1014b8	ce		into
0x1014b9	0f		
0x1014ba	b7	movzx ecx,WORD PTR	movzx ecx,WORD PTR
0x1014bb	48	[eax+0xa]	[eax+0xa]
0x1014bc	0a		
0x1014bd	83		
0x1014be	f1	xor ecx,0x7	xor ecx,0x7
0x1014bf	07		

Fazit

Kontrollflussfehler

Unausgerichtete Sprünge

Datenfehler



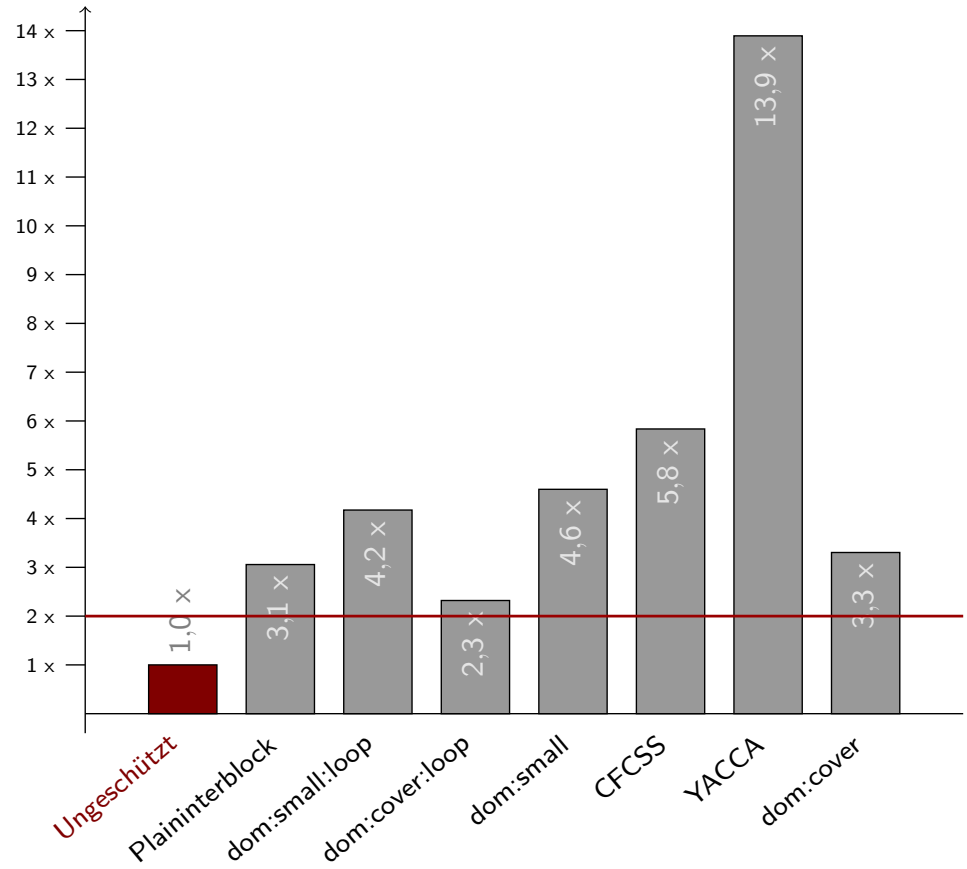
Fazit

Kontrollflussfehler

Unausgerichtete Sprünge

Datenfehler

Laufzeit



Referenzen

[1] M. Rebaudengo, M. S. Reorda, M. Torchiano und M. Violante, „Soft-error detection through software fault-tolerance techniques," in **Defect and Fault Tolerance in VLSI Systems, 1999. DFT'99. International Symposium on**. IEEE, 1999, S. 210–218.

[2] P. Cheynet, B. Nicolescu, R. Velazco, M. Rebaudengo, M. Sonza Reorda und M. Violante, „Experimentally evaluating an automatic approach for generating safety-critical software with respect to transient errors," **IEEE Transactions on Nuclear Science**, Vol. 47, Nr. 6, S. 2231–2236, 2000.

[3] Z. Alkhalifa, V. S. Nair, N. Krishnamurthy und J. A. Abraham, „Design and evaluation of system-level checks for on-line control flow error detection," **Parallel and Distributed Systems, IEEE Transactions on**, Vol. 10, Nr. 6, S. 627–641, 1999.

[4] N. Oh, P. Shirvani und E. McCluskey, „Control-flow checking by software signatures," **Reliability, IEEE Transactions on**, Vol. 51, Nr. 1, S. 111–122, Mar. 2002.

[5] O. Goloubeva, M. Rebaudengo, M. S. Reorda und M. Violante, „Soft-error detection using control flow assertions," in **Defect and Fault Tolerance in VLSI Systems, 2003. Proceedings. 18th IEEE International Symposium on**. IEEE, 2003, S. 581–588.

[6] O. Goloubeva, M. Rebaudengo, M. Reorda und M. Violante, **Software-implemented hardware fault tolerance**. Springer, 2006.

[7] C. Dietrich, „Global Optimization of Non Funcitonal Properties in OSEK Real-Time Systems by Static Cross-Kernel Flow Analyses," Masterarbeit, Universität Erlangen, Deutschland, Sep. 2014.