Static Checking of Interrupt-driven Software

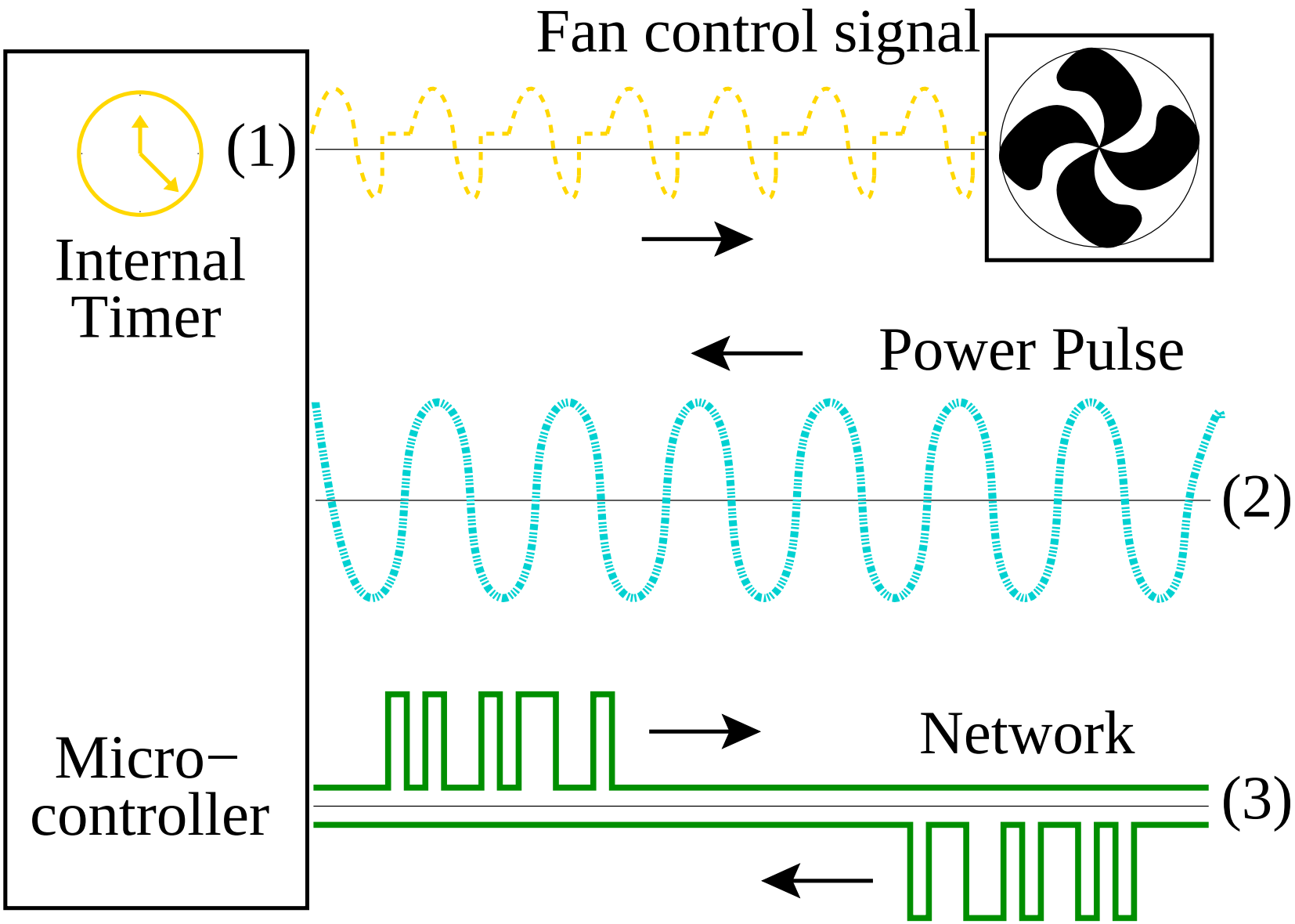
Jens Palsberg

Purdue University

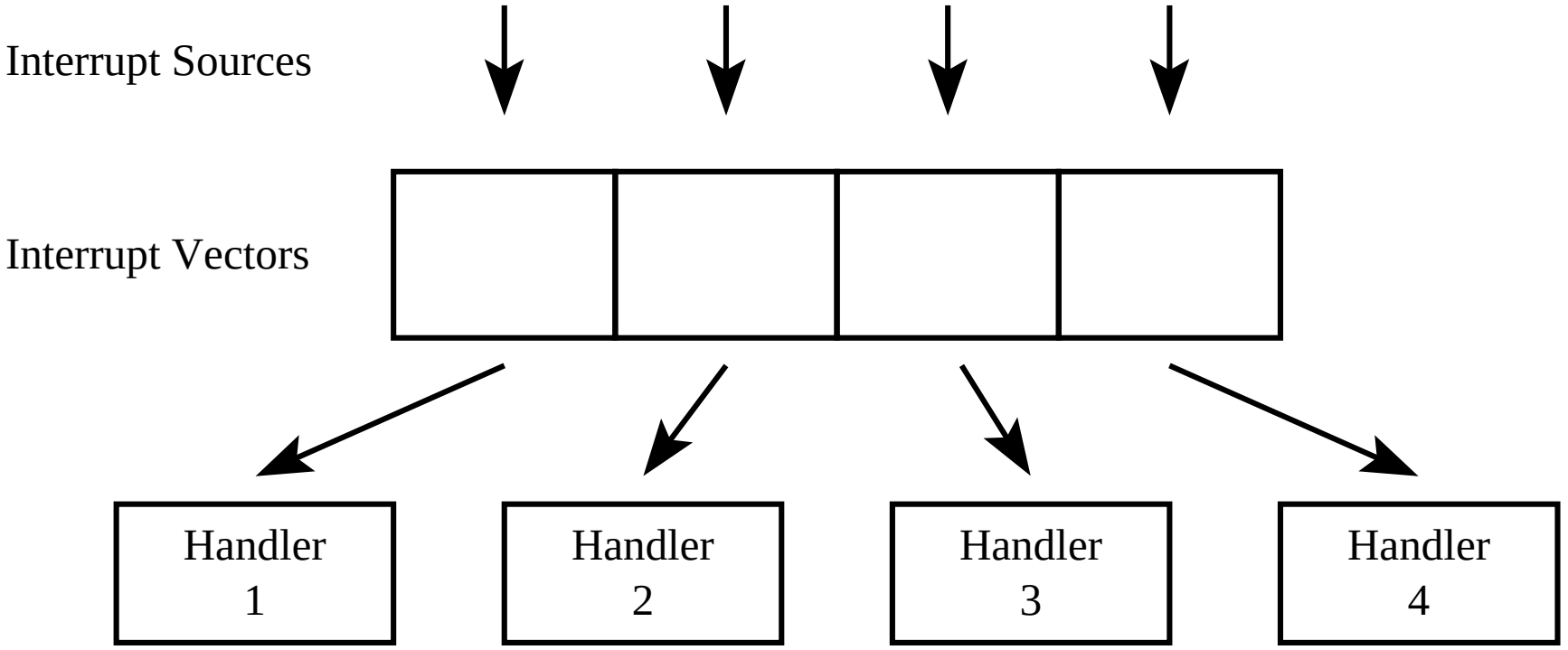
CERIAS and Department of Computer Science

www.cs.purdue.edu/people/palsberg

Joint work with Dennis Brylow and Niels Damgaard.
Supported by an NSF CAREER award.
Paper at International Conference on Software Engineering 2001.


Fan control signal
(1)
Internal
Timer
Power Pulse
(2)
Micro−
controller
Network
(3)

Interrupt-driven Control

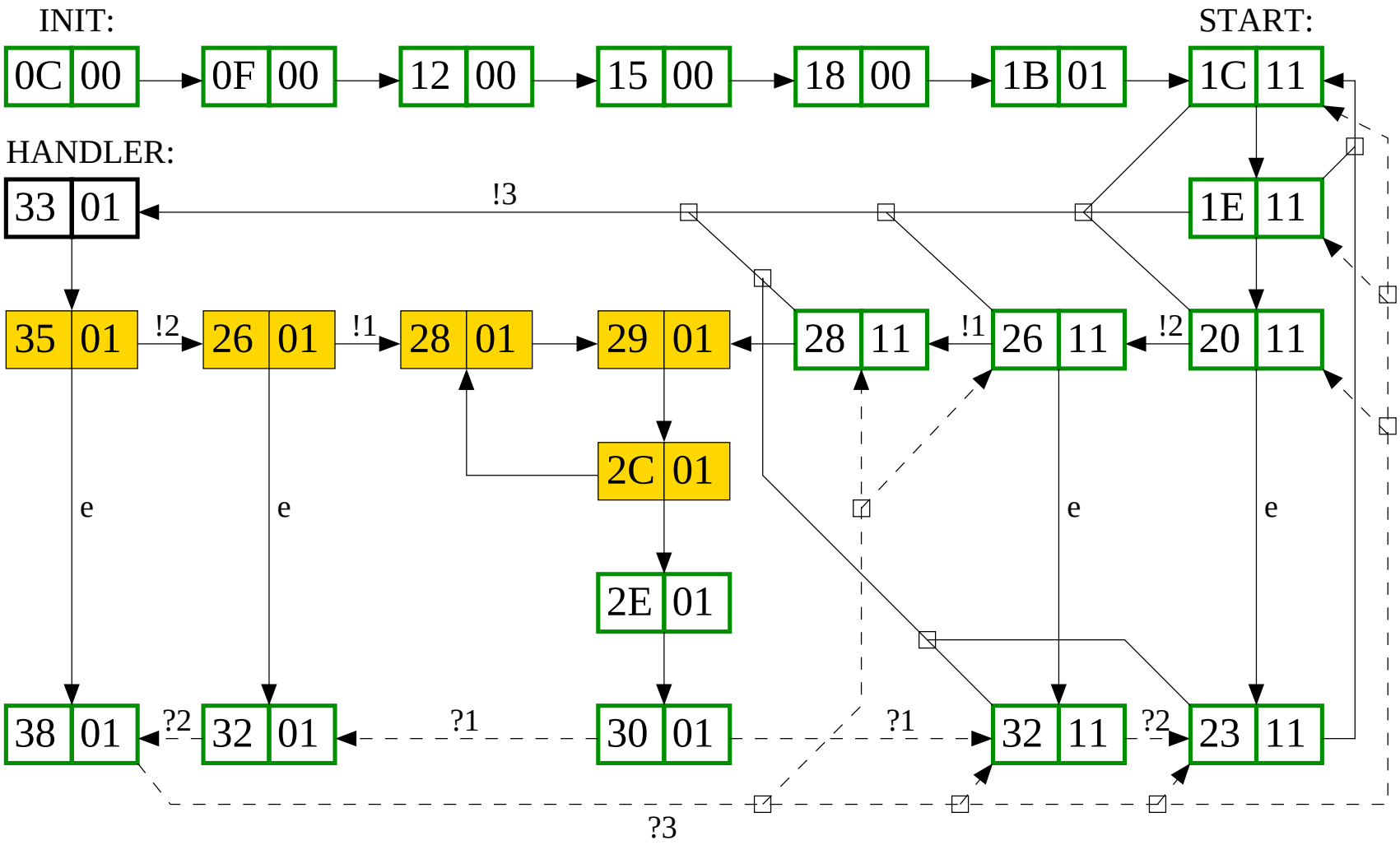


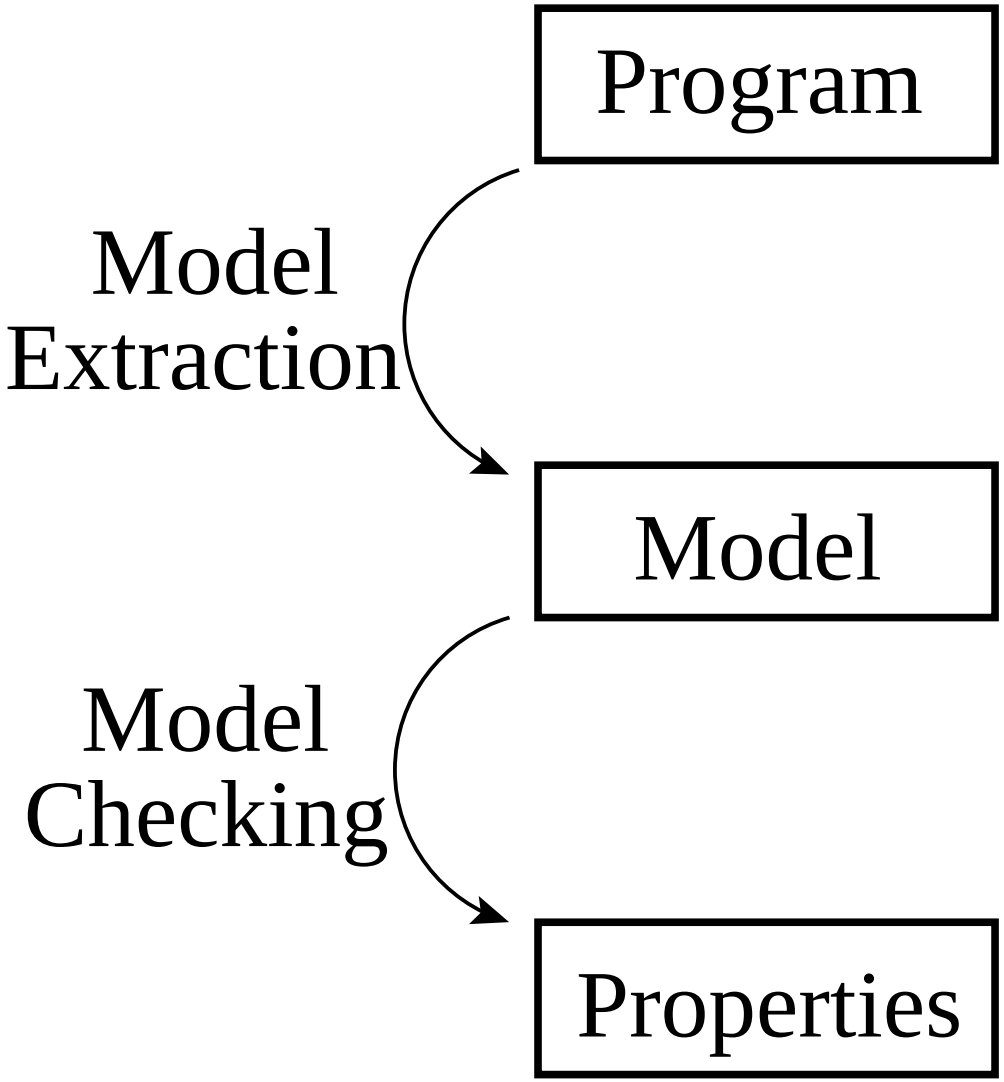
Example Program in Z86 Assembly Language

```
; Constant Pool (Symbol Table).
; Bit Flags for IMR and IRQ.
IRQ0 .EQU  #00000001b
; Bit Flags for external devices
; on Port 0 and Port 3.
DEV2 .EQU  #00010000b

; Interrupt Vectors.
     .ORG  %00h
     .WORD #HANDLER  ; Device 0

; Main Program Code.
     .ORG  0Ch
  INIT:              ; Initialization section.
0C   LD   SPL, #0F0h ; Initialize Stack Pointer.
0F   LD   RP,  #10h  ; Work in register bank 1.
12   LD   P2M, #00h  ; Set Port 2 lines to
                     ; all outputs.
15   LD   IRQ, #00h  ; Clear IRQ.
18   LD   IMR, #IRQ0
1B   EI              ; Enable Interrupt 0.
```

Example Program in Z86 Assembly Language

```
  START:                ; Start of main program loop.
1C   DJNZ r2,  START    ; If our counter expires,
1E   LD   r1,  P3       ; send this sensor's reading
20   CALL SEND          ; to the output device.
23   JP   START

  SEND:                 ; Send Data to Device 2.
26   PUSH IMR           ; Remember what IMR was.
  DELAY:
28   DI                 ; Musn't be interrupted
                        ; during pulse.
29   LD   P0,  #DEV2    ; Select control line
                        ; for Device 2.
2C   DJNZ r3,  DELAY    ; Short delay.
2E   CLR  P0
30   POP  IMR           ; Reactivate interrupts.
32   RET

  HANDLER:              ; Interrupt for Device 0.
33   LD   r2,  #00h     ; Reset counter in main loop.
35   CALL SEND
38   IRET               ; Interrupt Handler is done.
  .END
```


INIT:
START:
HANDLER:
0C 00
0F 00
12 00
15 00
18 00
1B 01
1C 11
33 01
!3
1E 11
35 01
!2
26 01
!1
28 01
29 01
28 11
!1
26 11
!2
20 11
2C 01
e
e
e
e
2E 01
38 01
?2
32 01
?1
30 01
?1
32 11
?2
23 11
?3


Program
Model Extraction
Model
Model Checking
Properties

Stack-Size Analysis

Program	Lower	Upper	Time	Space
CTurk	17	18	4.11 s	31.6 MB
GTurk	16	17	4.31 s	32.2 MB
ZTurk	16	17	4.22 s	32.1 MB
DRop	12	14	4.14 s	31.1 MB
Rop	12	14	4.18 s	31.8 MB
Fan	11	N/A	N/A	N/A
Serial	10	10	3.87 s	31.0 MB
Example	37	37	3.21 s	34.9 MB

The lower bounds were found with a software simulator for Z86 assembly language that we wrote.

Interrupt Latency Analysis of the Highest Priority IRQ

Program	Green	Yellow	Red	Latency
CTurk	51%	49%	0%	260
GTurk	50%	50%	0%	272
ZTurk	50%	50%	0%	276
DRop	19%	81%	0%	312
Rop	19%	81%	0%	312
Fan	67%	33%	0%	310
Serial	79%	21%	0%	326
Example	46%	54%	0%	242

Latencies are given in machine cycles.

One machine cycle is executed in 1 microsecond.

Secure Software Systems Group

Department of Computer Science

http://www.cs.purdue.edu/s3/

Faculty: Antony Hosking, Jens Palsberg, Jan Vitek.

16 Ph.D. students, 2 M.S. students, 3 undergraduate students.

Sample projects: Java security, bytecode compression, interoperability of software systems, real-time system verification, software watermarking, high-performance persistent object storage.

Funding: NSF, DARPA, CERIAS, Sun Microsystems, Lockheed Martin, IBM, Motorola, and Intel.