Software/Hardware Integration

LEGO RCX
Hitachi H8/3292

LISHA/UFSC

Fauze Valério Polpeta

Prof. Dr. Antônio Augusto Fröhlich

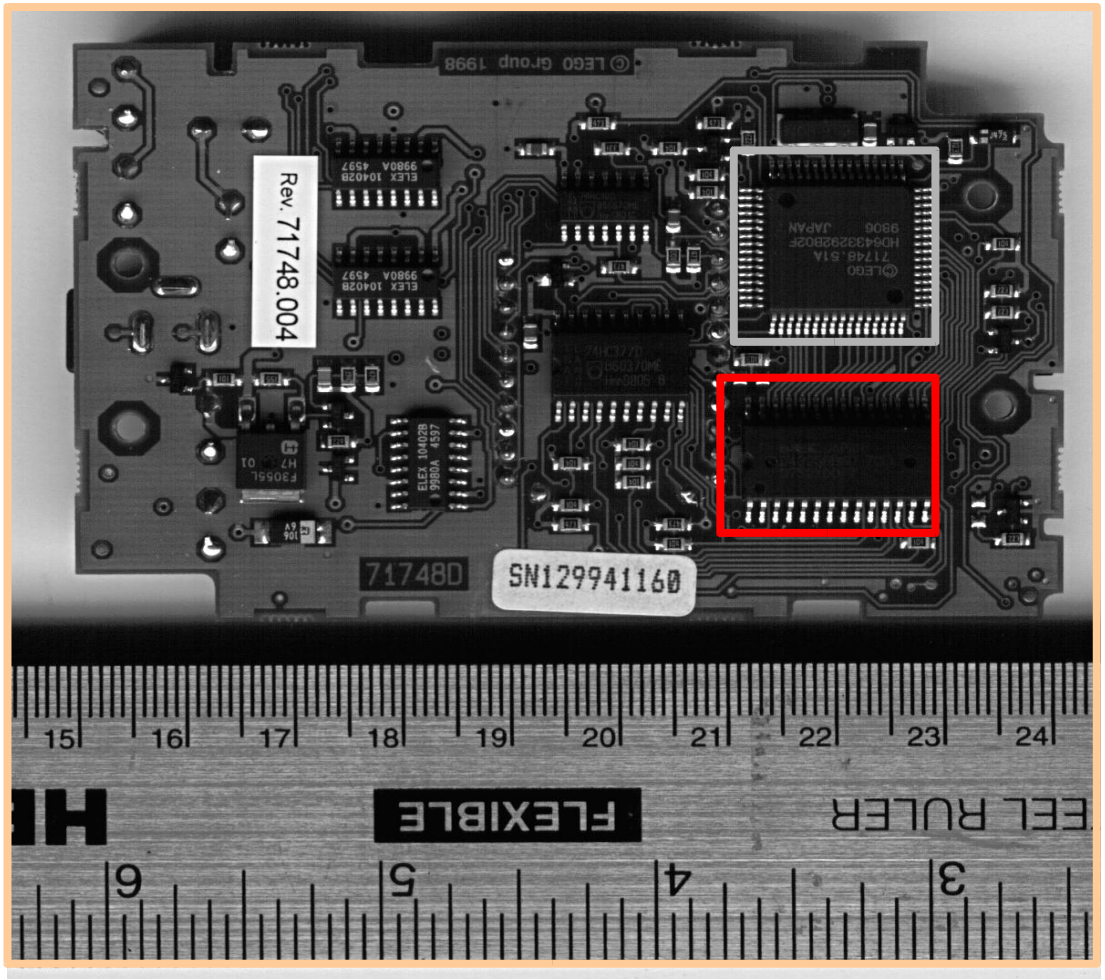
{fauze|guto}@lisha.ufsc.br

http://www.lisha.ufsc.br/~{fauze|guto}

March 2003

LEGO RCX Overview

- LEGO RCX
  - Programmable hardware module
  - Interface to I/O devices (sensors and actuators)
  - Hitachi H8/3292 microcontroller

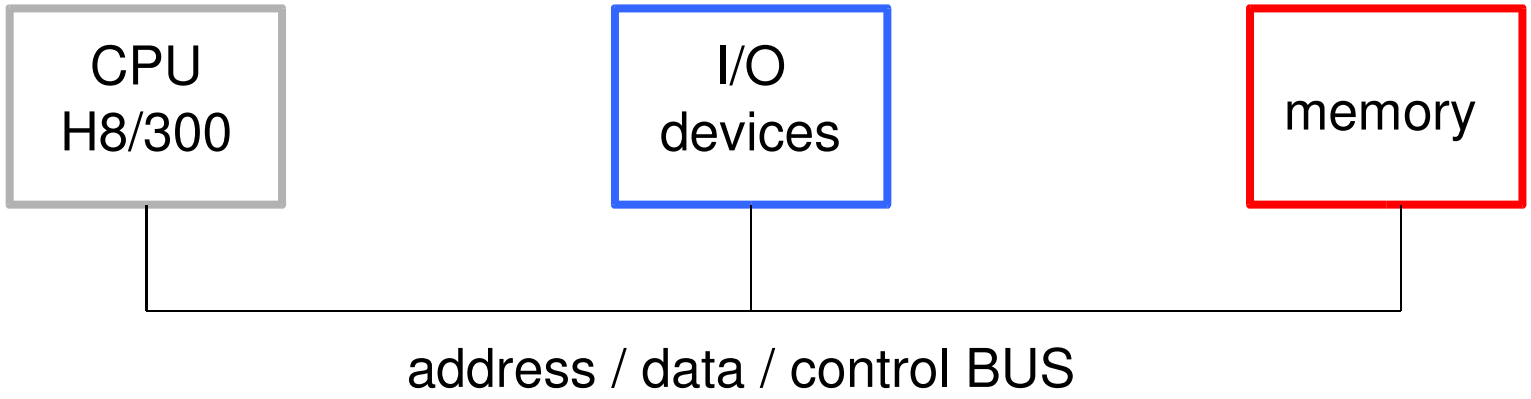


LEGO RCX I/O Devices

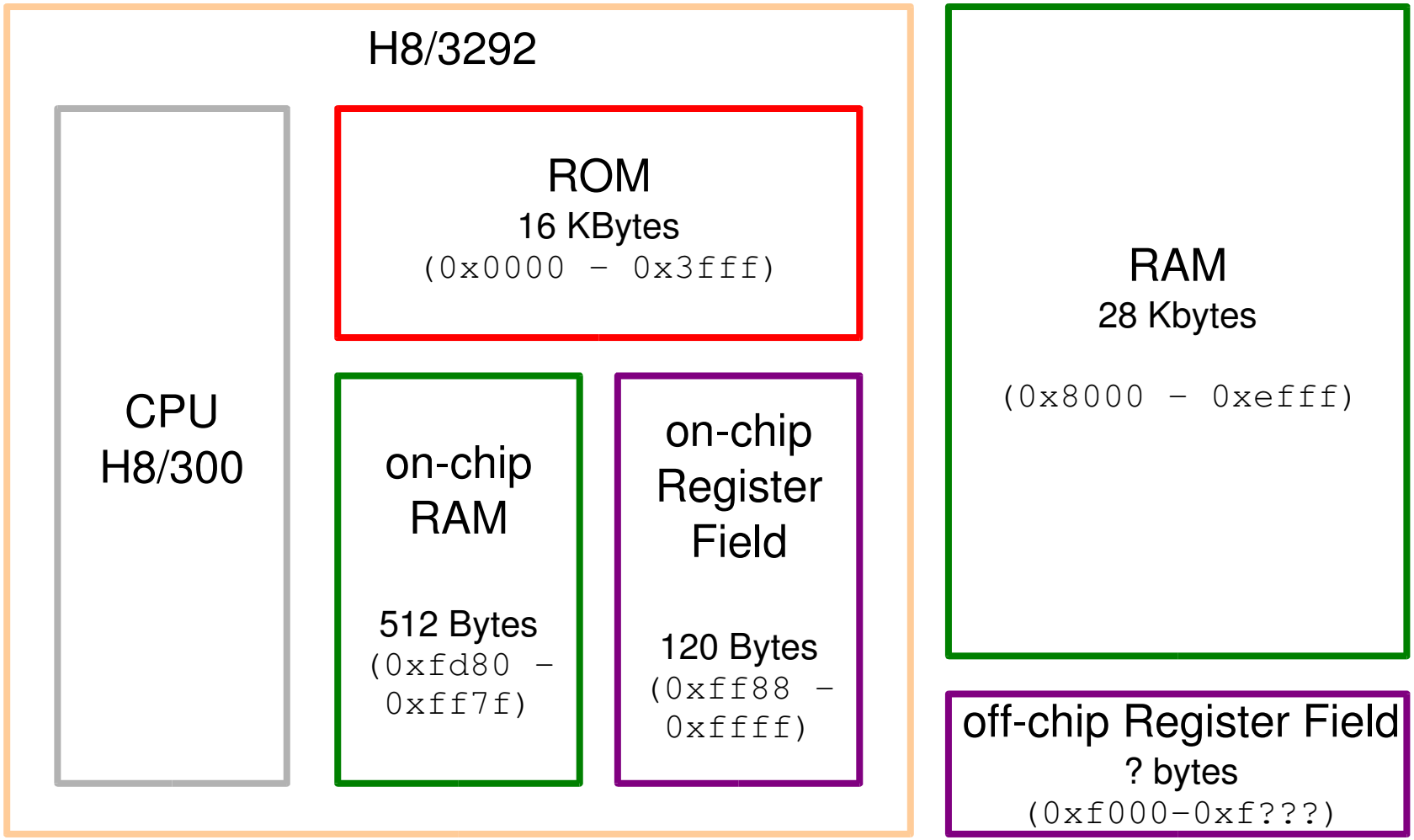
- Input devices
  - 3 sensor ports (through A/D converters)
  - 4 buttons
  - battery level monitor
  - timers
- Output devices
  - 3 actuator ports (through D/A converters)
  - 5-segment LCD
  - speaker
- Bidirectional devices
  - infrared port

Hitachi H8/300 Overview

- H8/300 CPU
  - 8-bit data
  - 16-bit address space
  - 8 x 16 bit GP registers (r0 - r7)
    - r0 => function return
    - r7 => stack pointer
  - 16 MHz clock



Hitachi H8/3292 Block Diagram

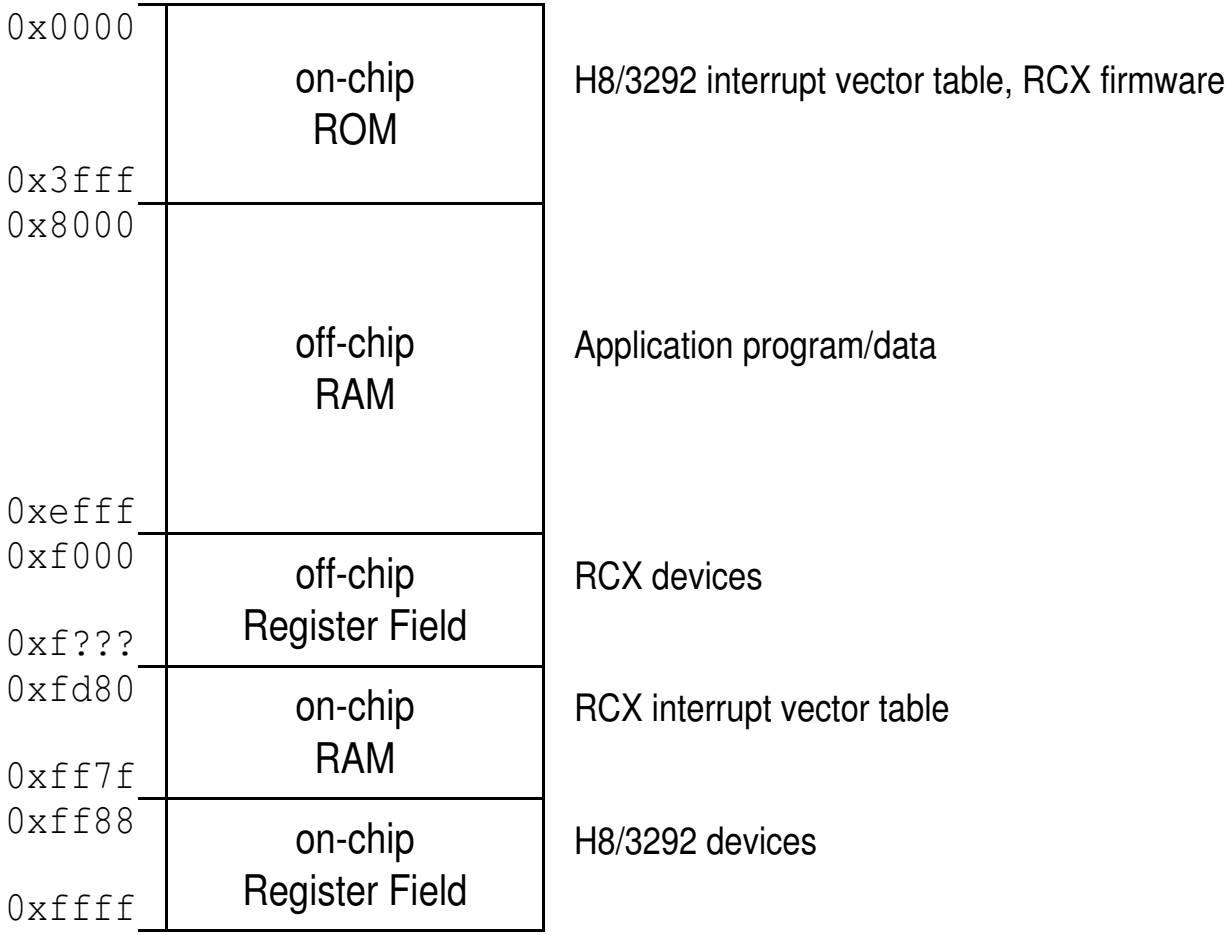


Hitachi H8/300 Address Modes

- Address modes
  - Register direct: rn
  - Register indirect: @rn
  - Register indirect with 16-bit displacement: @(d:16,rn)
  - Register indirect with post-increment: @rn+
  - Register indirect with pre-increment: @-rn
  - Absolute address (8 or 16 bits): @aa:8, @aa:16
  - Immediate (8 or 16-bit data): #aa:8, #aa:16
  - PC-Relative (8-bit displacement): @(d:pc)
  - Memory indirect: @@aa:8

LEGO RCX Memory Layout

- MCU mode 2 (control register at `0xffc5`)

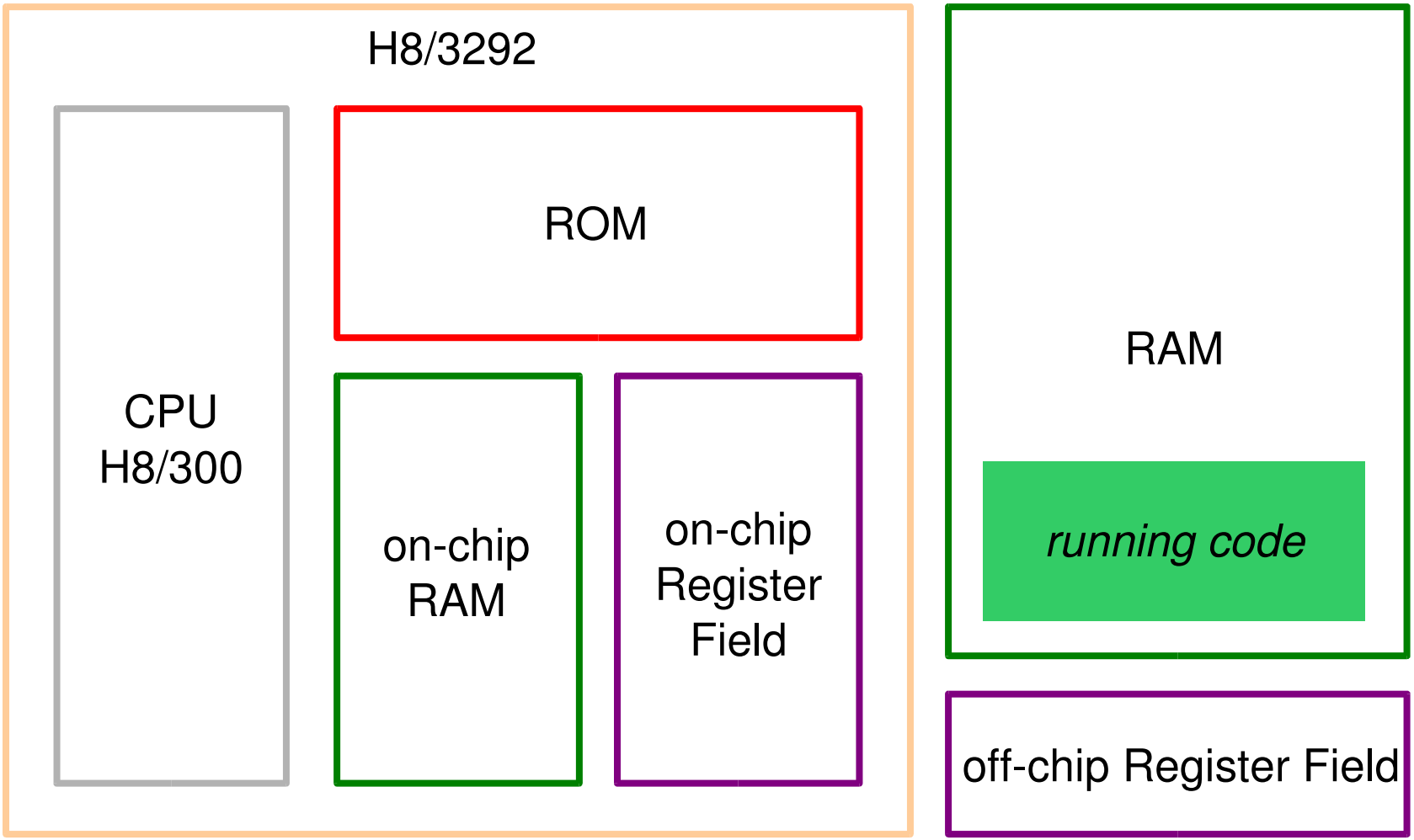


LEGO RCX Interrupt Dispatching

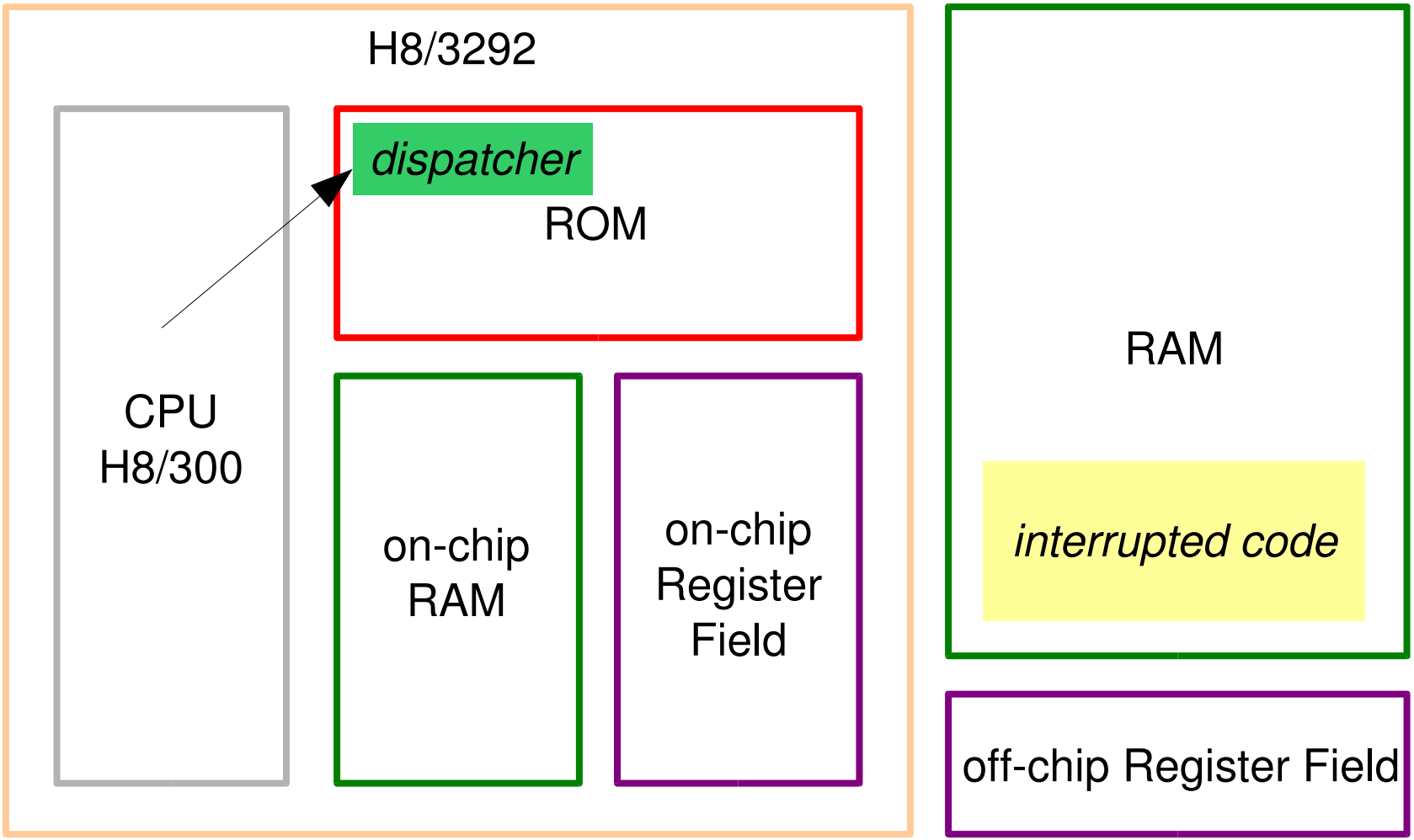
- H8/3292 interrupt table
  - Stored at `0x0000-0x0049` (ROM in the RCX)
  - RCX ROM vectors redirect interrupts to the on-chip RAM interrupt table
  - Decreasing priority
- on-chip RAM interrupt table
  - Stored at `0xfd80-0xfdbf`
  - Pointers to user-defined handlers
- Masking
  - Globally (except NMI) CCR bit 7
  - Individually through the off-chip register field

Vector	Source
0	reset
1 - 2	reserved
3	NMI
4 - 6	IRQs
7 - 11	reserved
12 - 18	16-bit timer
19 - 21	8-bit timer 0
22 - 24	8-bit timer 1
25 - 26	reserved
27 - 30	serial
31 - 34	reserved
35	ADI
36	WOVF

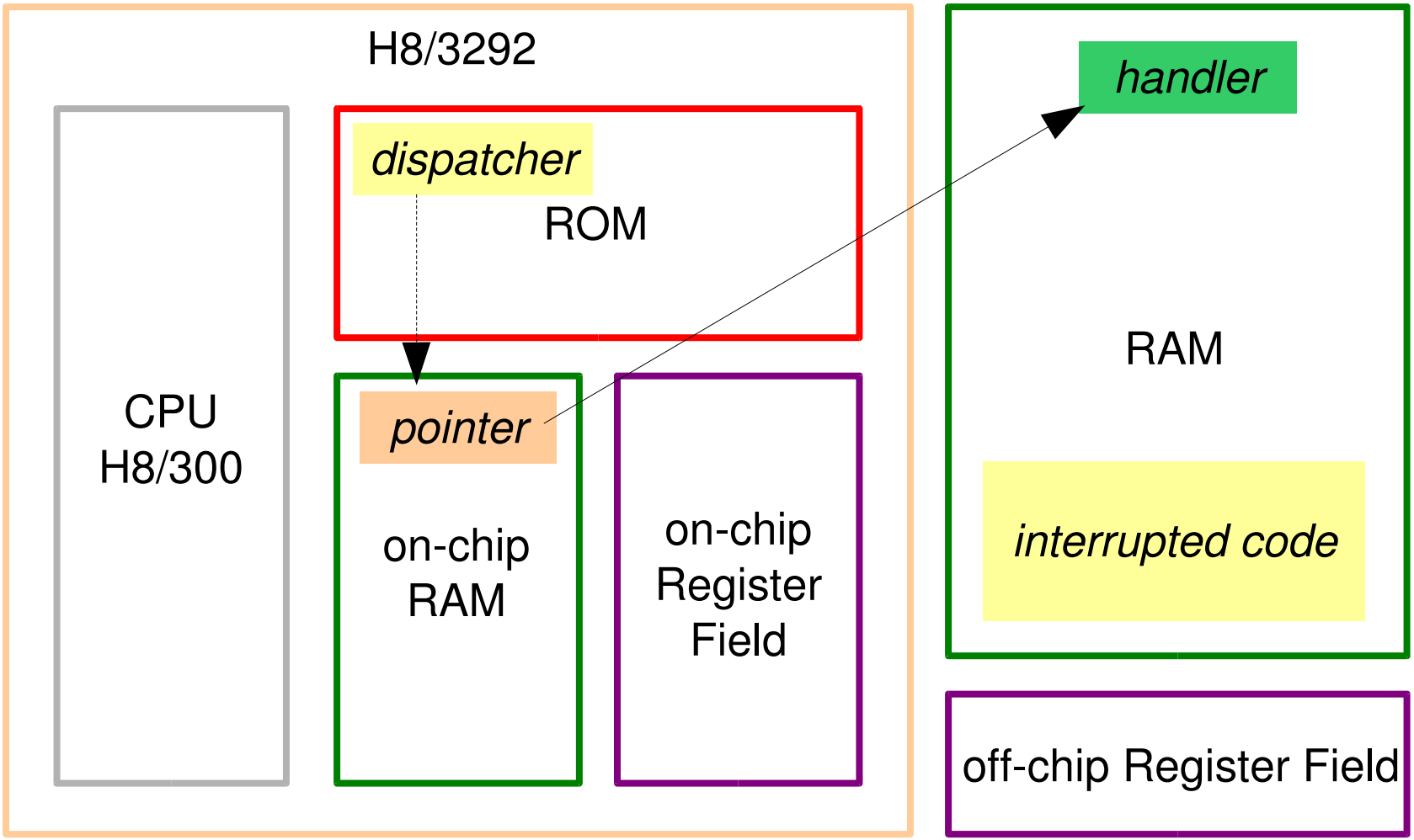
Interrupt Dispatching



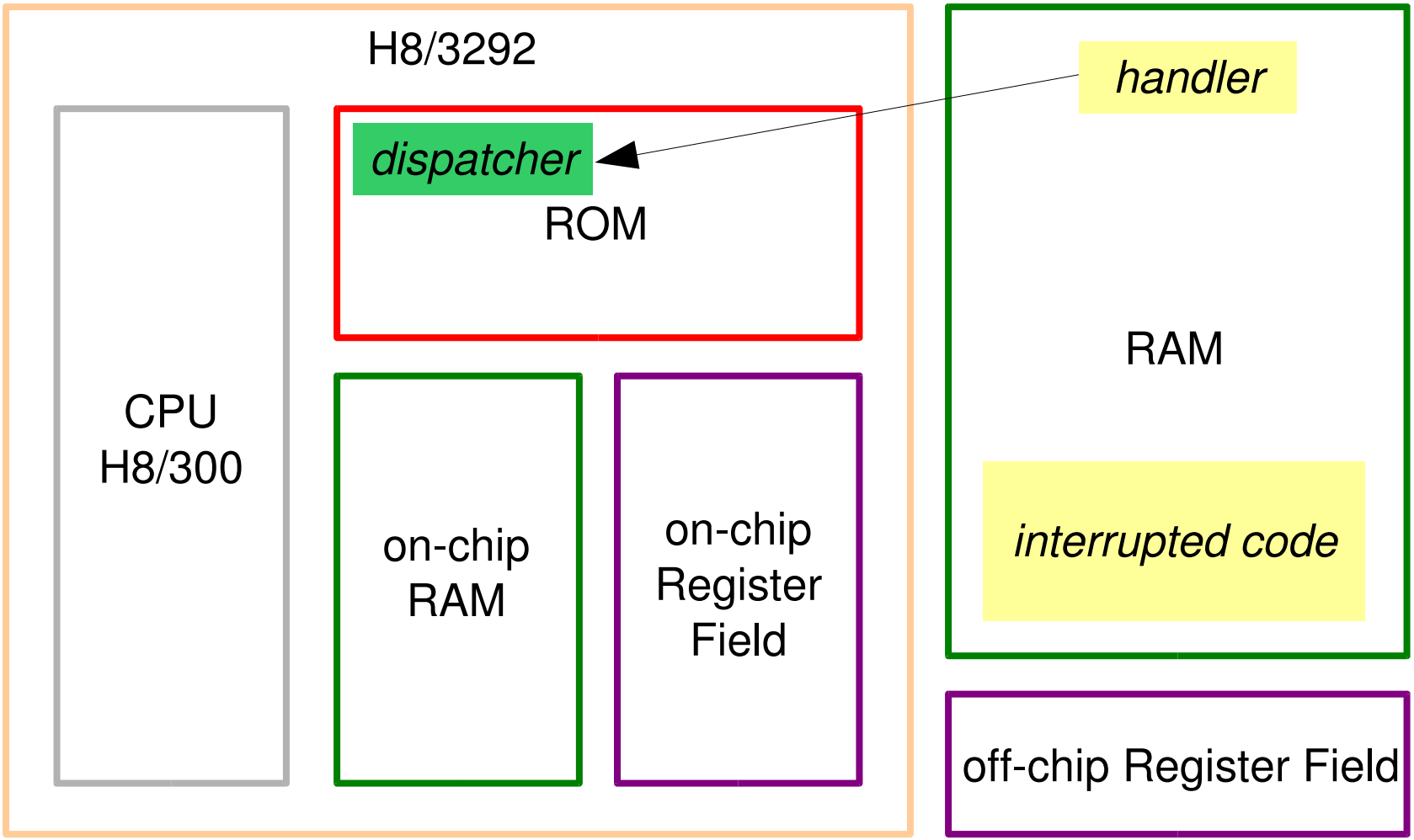
Interrupt Dispatching



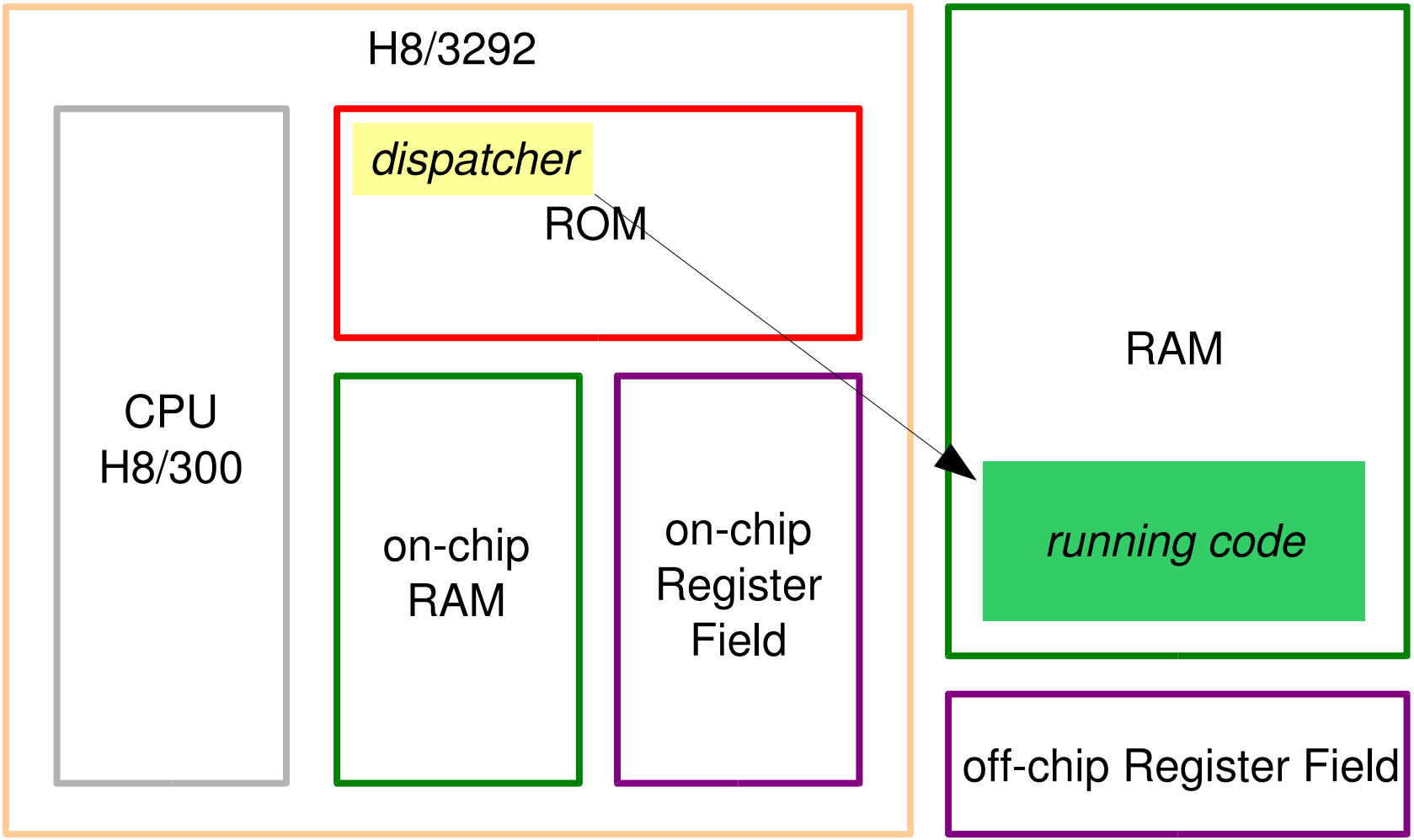
Interrupt Dispatching



Interrupt Dispatching



Interrupt Dispatching



LEGO RCX Interrupt Handling

- H8 dispatching

```
push pc
push ccr
ccr[7]=1 /* int disable */
H8_Int_Table[n]();
pop ccr
pop pc
```

- H8/300 Handler (ROM)

```
void h8_handler(void) {
  push r6
  mov  RCX_Int_Table[n],
  r6
  jsr  @r6
  pop  r6
  rte
};
```

- RCX Handler

```
void rcx_handler(void) {
  /* push registers */
  /* handle interrupt */
  /* pop registers */
};
```

- RCX Interrupt table

```
typedef void (RCX_Handler)(void);
RCX_Handler ** RCX_Int_Table = (RCX_Handler **)0xfd80;
RCX_Int_Table[n] = &rcx_handler;
```

Hitachi H8/3292 I/O

- 7 I/O ports
  - P1-P4,P6: 8-bit, input/output
  - P5: 3-bit, input/output
  - P7: 8-bit, input
- Each comprised of up to three registers
  - DDR (P1-P6): data direction (input/output)
  - DR (P1-P7): data
  - PCR (P1-P3): pull-up control
- 3 operating modes
  - I: I/O address space, on-chip ROM disabled
  - II: I/O address space, on-chip ROM enabled
  - III: general I/O

Hitachi H8/3292 I/O

Port	Type	Mode I	Mode II	Mode III	DDR	DR	PCR
1	8-bit, I/O	16-bit I/O address (lsb)		GPIO	0xffb0	0xffb2	0xffac
2	8-bit, I/O	16-bit I/O address (msb)		GPIO	0xffb1	0xffb3	0xffad
3	8-bit, I/O	8-bit I/O data		GPIO	0xffb4	0xffb6	0xffae
4	8-bit, I/O	bus state (IRQ/WAIT/RD/WR/CLOCK/AD)		GPIO	0xffb5	0xffb7	–
5	3-bit, I/O	serial port			0xffb8	0xffba	–
6	8-bit, I/O	timer control			0xffb9	0xffbb	–
7	8-bit, I	A/D converter			–	0xffbe	–

Hitachi H8/3292 I/O

- A/D converter
  - Control/status register to initiate and monitor conversions
  - Can trigger interrupts

ADCSR at `0xffe8`

Bit	Name	Abbreviation	Function
7	A/D End Flag	ADF	1 = end of conversion(must be cleared)
6	A/D End Interrupt Enable	ADIE	0 = disabled; 1= enabled
5	A/D Start	ADST	0 = stop; 1 = start
4	Scan Mode	SCAN	0 = single; 1= scan
3	Clock Select	CKS	0 = slow; 1= fast
2 - 0	Channel Select	CH2 - CH0	`000` CH0, `001` CH1, `010` CH2, `011` CH3

Hitachi H8/3292 I/O

- A/D converter
  - Three 10-bit channels (ADDR[A|B|C)

ADDR

Channel	Data Reg.	Addr.
AN0	A	`0xffe0`
AN1	B	`0xffe2`
AN2	C	`0xffe4`

LEGO RCX I/O

- H8/3292 I/O ports in operating mode II
- Buttons
  - 4 buttons connected to H8 I/O ports 4 and 7
  - `Run` and `OnOff` buttons can trigger interrupts

RCX Button	H8 Port/Bit	H8 IRQ
Run	4/2	0
OnOff	4/1	1
View	7/6	-
Prgm	7/7	-

LEGO RCX I/O

- Sensors
  - 3 ports (1, 2, 3)
  - Connected to H8 I/O port 7 (A/D converter)
  - Passive (e.g. touch, temperature)
  - Active (e.g. light, rotation)
    - Activation via H8 I/O port 6

RCX Sensor	H8 P7 bit (A/D ch)	H8 P6 bit (activate)
0	AN2	2
1	AN1	1
2	AN0	0

RCX Executive

- After a power up or reset, the H8/300 uses the first entry in the interrupt vector table (address `0x0000`) to determine the starting address
- In the RCX, this procedure activates an executive that
  - Monitors the infrared port for application uploads
  - Builds up a run-time environment for application programs, with services such as infrared communication, sensor and actuator manipulation, etc