

SOFTWARE - ØVELSE 3

REALTID, PRU

E4ISD2

Janus Bo Andersen (JA67494)

ØVELSE 3: REALTIDSPROCESSERING DESIGN

Formål: Afdæk hvor hurtigt et firkantsignal kan propagere gennem PRU'erne, så de "stadic kan følge med".

Design:

- Målrettet at opnå lav latency gennem PRU'erne:
- PRU0 aflæser GPIO-input direkte
- Scratchpad (SPAD) benyttes til at overføre data fra PRU0 til PRU1
 - XOUT / XIN operationer
- PRU1 sætter GPIO-output direkte
- Kode skrevet/optimeret i assembler.

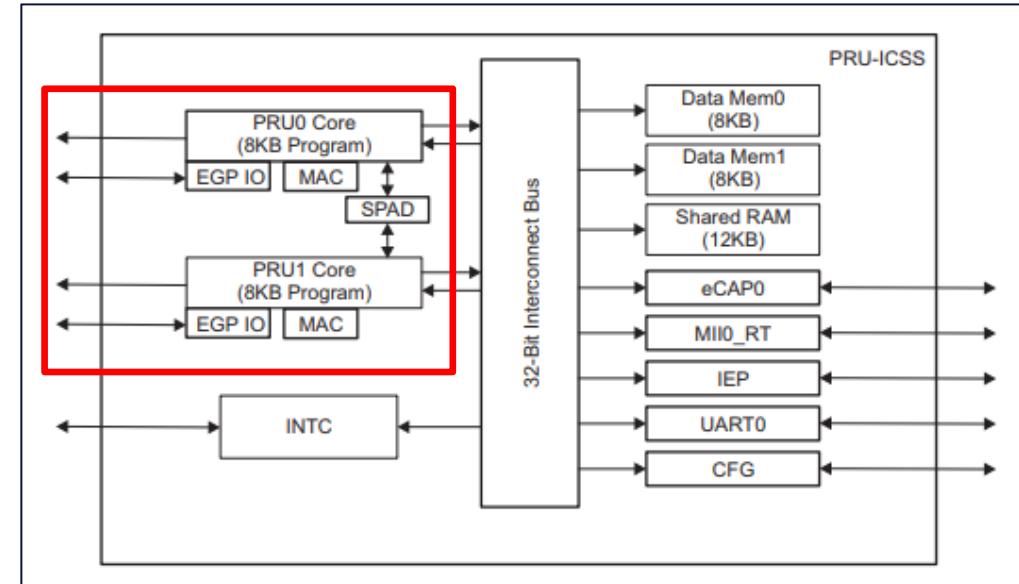
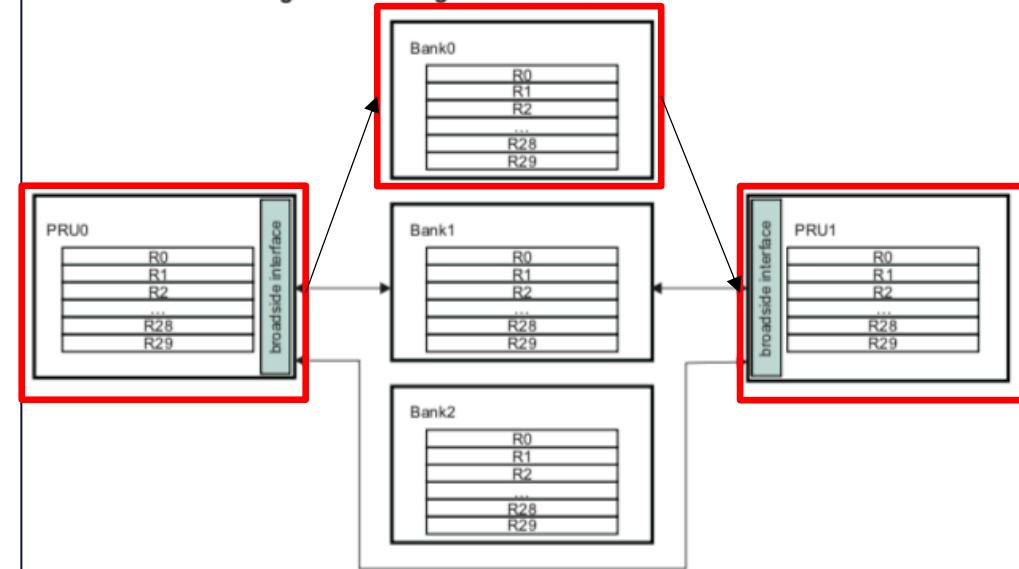
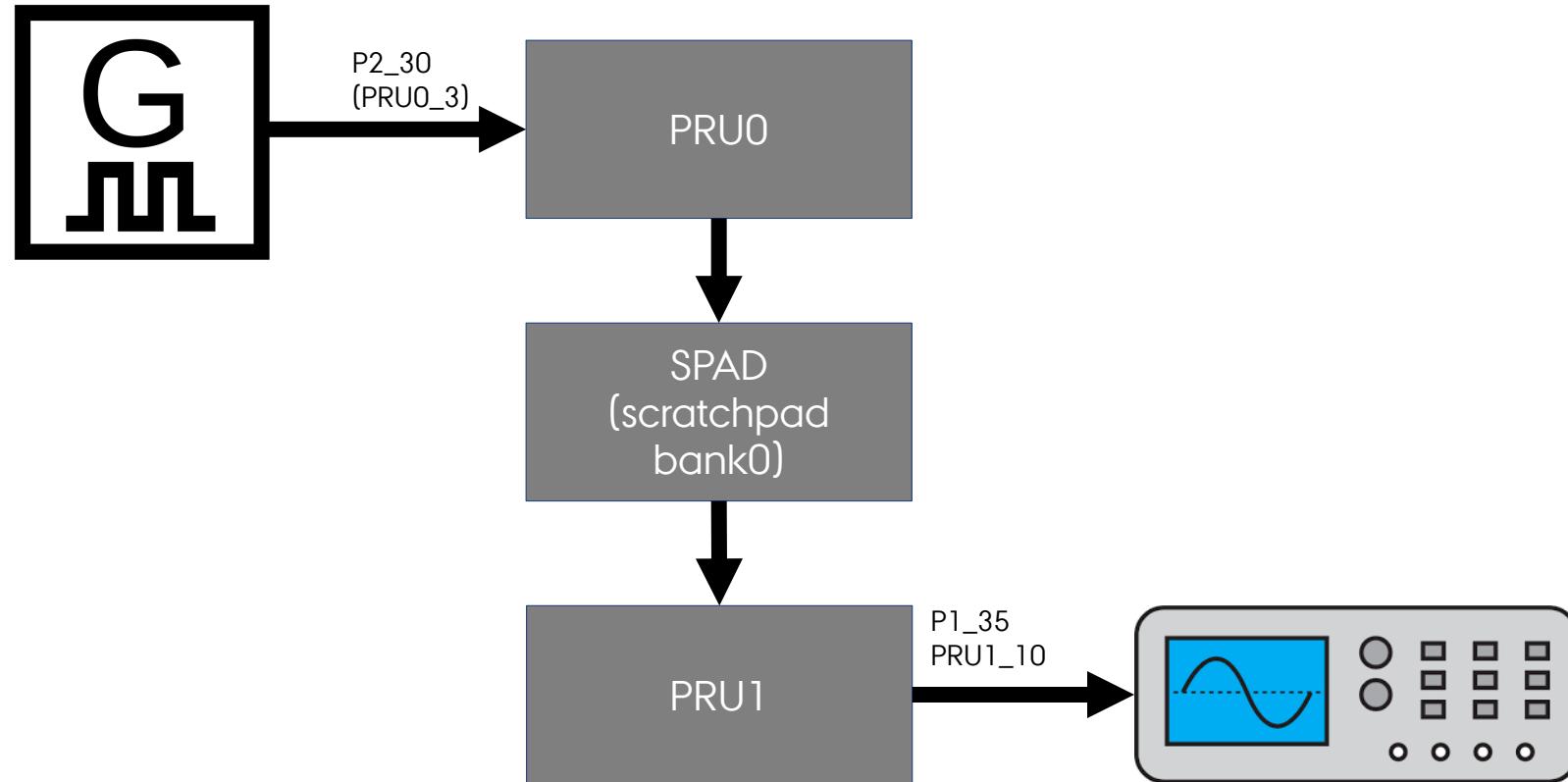


Figure 4-16. Integration of PRU and Scratch Pad



BLOKDIAGRAM

FIRKANTINPUT PÅ PRU0, OUTPUT FRA PRU1



TESTSETUP

Pins for PRU0 og PRU1 sættes op til hhv. in- og output

```
config_pru_pins.sh
1 # input pin PRU0_3, i.e. pr1_pru0_r31_3
1 config-pin P2_30 pruin
2
3 # output pin PRU1_10, i.e. pr1_pru1_r30_10
4 config-pin P1_35 pruout
```

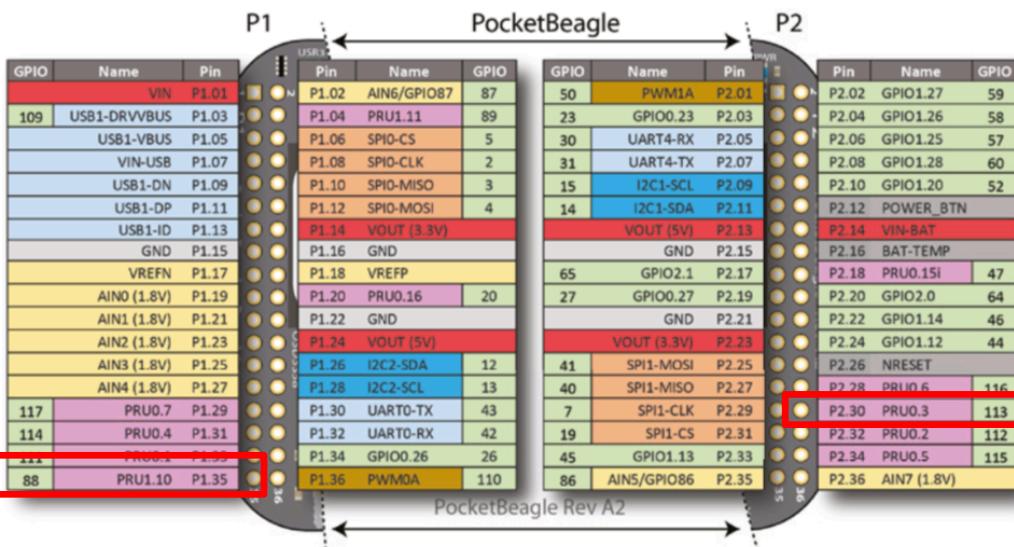
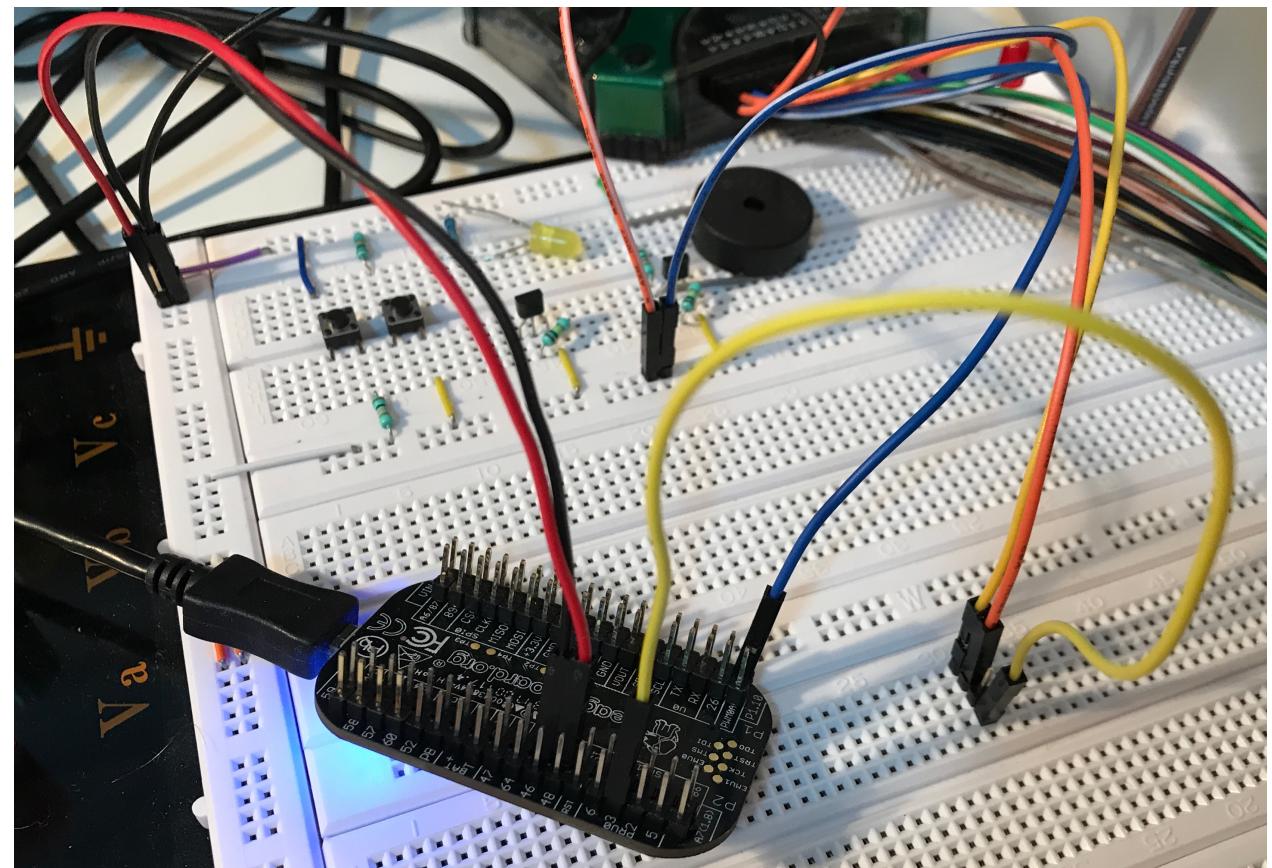


Figure 6-2: The PocketBeagle P1/P2 headers with pin names, which describe each pin's default functionality



Elektrisk:

- Fælles stelplan for PocketBeagle og Analog Discovery.
- Wavegen1 (gul) og oscilloskop kanal 1 (orange) ved P2_30 (pruin).
- Oscilloskop kanal 2 (blå) ved P1_35 (pruout).

ASSEMBLER

Instruktioner til PRU0

```
1      .cdecls "perfTestContainer.c"
2
3      .clink
4      .global start
5      start: CLR    r30, r30.t5      ; turn off the LED
6      SET    r0, r0.t10      ; Initialize value to send to PRU1
7      CHECK: WBS    r31, 3       ; wait for bit set on GPIO in
8      XOUT   10, &r0, 4       ; Send value to PRU1
9      CLR    r0, r0.t10      ; Prepare the next value to send to PRU1
10     WBC    r31, 3       ; wait until the bit is cleared again
11     XOUT   10, &r0, 4       ; Send value to PRU1
12     SET    r0, r0.t10      ; Prepare value for next round to PRU1
13     JMP    CHECK        ; Repeat
14
15     HALT
```

WBS og WBC er
pseudoinstruktioner, der
indefholder loop og compare
(busy polling)

Afsender besked via SPAD så
hurtigt som muligt, og
forbereder registeropdatering
til næste "edge" bagefter.

Instruktioner til PRU1

```
1      .cdecls "perfTestContainer.c"
2
3      .clink
4      .global start
5      start: CLR    r30, r30.t10      ; start with GPIO low
6      CHECK: XIN    10, &r0, 4       ; read from scratchpad bank 0
7      MOV    r30, r0        ; Move value from scratchpad into GPIO out
8      JMP    CHECK        ; loop
9
10     HALT
```

Aflæser kontinuerligt
scratchpad, og indsætter
værdien i GPIO out registeret.

Det kunne være smukt at
advisere PRU1 via interrupt
om at værdien er opdateret,
men dette her virker
umiddelbart hurtigt

PERFORMANCE TEST - 1 PULS

Kanal 1: Input til PRU0 (knap)

Kanal 2: Output fra PRU1

- 40 ns fra input høj til output høj
- => 8 clock cycles @200 MHz

Regnskab over ticks:

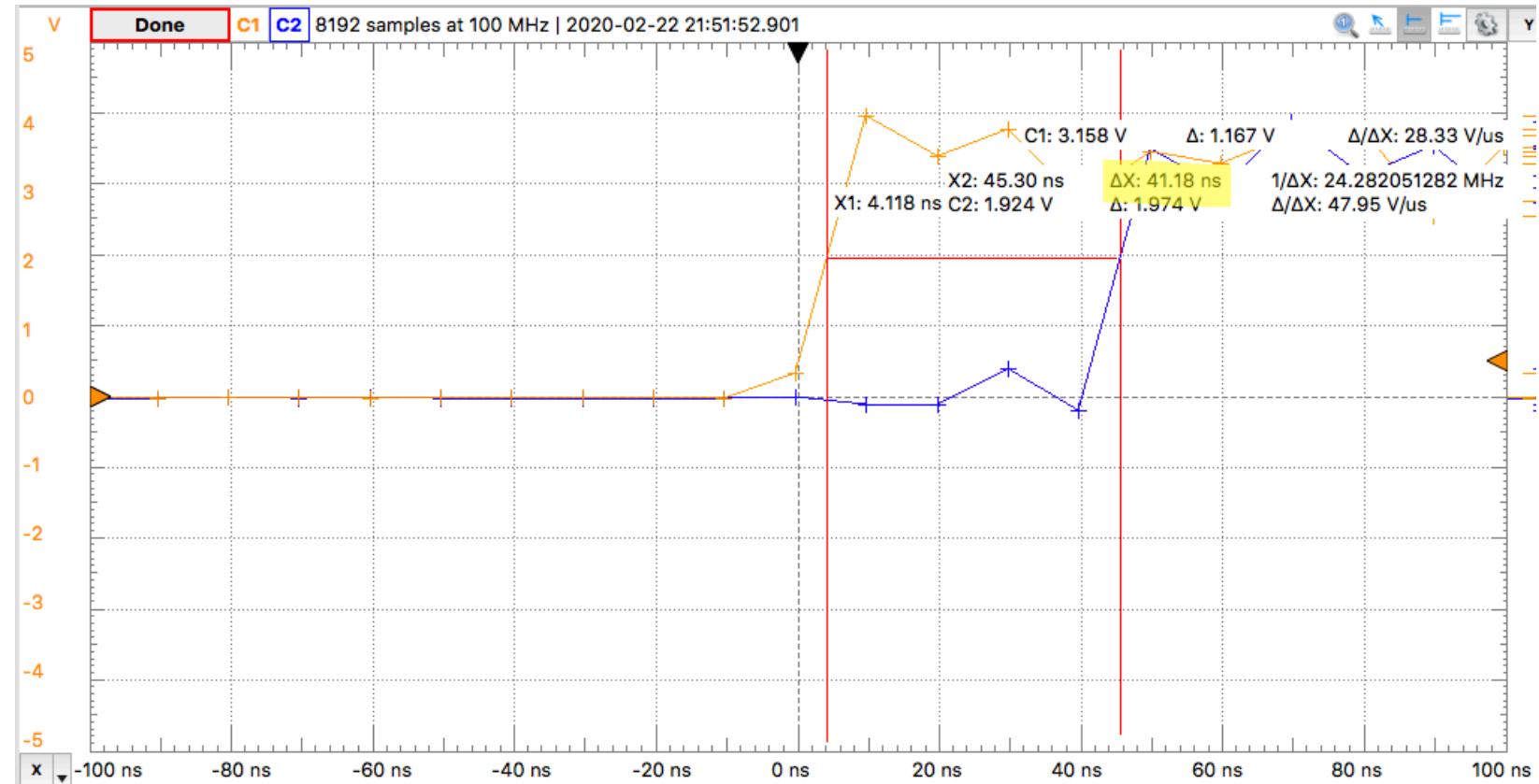
- Propagation delay/internal delay?

PRU0 latency:

- 3-4 ticks WBS/WBC-'spinning' indtil GPIO aflæst og registreretændret
- 1 tick til at overføre værdi til SPAD

PRU1 latency:

- 2-3 ticks i loop til at læse SPAD, opdatere GPIO out, JMP



PERFORMANCE TEST - WAVE GENERATOR

Setup: Se t.h.

- Definition:
- **OK latency: ≤ 50 ns**

Frekvens på firkantsignalet øges:

- Se næste billeder
- OK op til RF-frekvenser
- Herefter giver det ikke mening at måle mere



LATENCY VED 1 MHZ FIRKANTSIGNAL

Ved 1 MHz ligger latency
stadic omkring 40 ns

PRU kan stadic følge med!

I nærheden af HF-delen i RF-
båndet...

- Ikke-ideelle effekter begynder at spille ind
 - Selv-induktive effekter i ledninger
 - Kapacitans i breadboard, m.m.



LATENCY VED 5 MHZ FIRKANTSIGNAL

Latency gennem PRU'erne stadig
< 50 ns ved 5 MHz:

- Signalkvalitet ind er væsentligt forringet
- Kraftig ringning på signal ud

Stopper forsøget her, og konkluderer, at måleudstyret er outperformed af PRU'erne

- Skal benytte BNC-prober for at komme det nærmere.



KONKLUSION

Pga. meget **lav** og **deterministisk** (worst-case) **latency**, er PRU'erne velegnede til:

- Hård-realtime:
 - Garanterede deadlines vha. deterministisk afvikling ved 200 MHz / 5 ns ticks.
- Håndtere højfrekvent input/output (i hvert fald op til HF), fx
 - Afstandsmåling, timing, osv.
 - Stabil/præcis signalgenerator.
- Afkoble og synkronisere behandling af signaler
 - Input-signal til én PRU og output-signaler fra en anden
 - Synkronisering via delte memory-områder med meget lave read latencies (1 tick? for SPAD, 3 for Shared DRAM).
- Arbejde i realtid uafhængigt af Linux-kernen.
- Synkronisere via beskeder til/fra Linuxkernen vha. remoteproc.

APPENDIKS: BEST-CASE LATENCY (PRU LOKALT)

A.1 AM335x

Table 1 through Table 3 are considered "best-case" read latency values for the PRU on AM335x.

Table 1. AM335x PRU Read Latencies - Local PRU Subsystem Resources

MMRs	Read Latency (PRU cycles @ 200 MHz)
PRU CTRL	4
PRU CFG	3
PRU INTC	3
PRU DRAM	3
PRU Shared DRAM	3
PRU ECAP	4
PRU UART	14
PRU IEP	12
PRU R31 (GPI)	1

<http://www.ti.com/lit/an/sprace8a/sprace8a.pdf>