

Laptop changes

Tuesday, 25 January 2022 18:05

- ☐ Ändra följande rader i /etc/sysctl.conf:

```
net.core.rmem_max=26214400
net.core.rmem_default=26214400
```

Alternativt, kör följande kommandon efter varje boot:

```
sysctl -w net.core.rmem_max=26214400
sysctl -w net.core.rmem_default=26214400
```

From <https://www.google.com/search?q=Increase+size+of+udp+buffers+in+Linux&rlz=1C1BNSD_enSE941SE941&oq=Increase+size+of+udp+buffers+in+Linux&aqs=chrome..69i57j33i22i29i30i6.11265j0j4&sourceid=chrome&ie=UTF-8>

From <<https://askubuntu.com/questions/1115636/how-do-i-increase-the-udp-receive-buffer-size>>

Även med ovanstående visade det sig att enstaka paket kunde förkomma. Det är inte säkert att det är laptopens fel, men jag satte även net.ipv4.udp_rmem_min (Minsta garanterad bufferstorlek per UDP-port) till ett större värde (c:a 1MB) och sedan dess har jag inte haft problem med borttappade frames.

- ☐ Aktivera Jumbo frames ? Verkar inte behövas? Jag har inte gjort något på laptopen.
- ☐ Nyare Linuxar har döpt ethernet-interfacet till enp3s0 (istf eth0). Ändringar krävs i vissa script och kommandon som beskrivs i testinstruktionerna.
- ☐ Jag gjorde ett script i ~/wdir/ilcdaq/config-v10.0/power som kör kommandon till 5-to-1-kortet. Stäng putty och kör detta script istället. Det initialiserar I2C och IO-expanders, gör ROA00 och väntar sedan på att man slår igång 3.1V. När man sedan trycker på 'Enter' gör den resten av initialiseringen (ROA06, DIN, osv). Sedan kan man köra egna UART-kommandon. Hoppa ur scriptet genom att trycka 'q'. Det är lätt att ändra scriptet till att skicka andra kommandon.
- ☐ Två script "serialize" och "rampcheck" i data-katalogen bearbetar data som kommer ur sruunpack. Serialize matar ut en fil med en rad per event, kanal och sampel (istället för tre per rad som kommer ur sruunpack). Rampcheck tar denna serialiserade fil och kollar att det är en uppåtgående eller neråtgående ramp. Avvikelser rapporteras. Prova med t.ex. "sruunpack -d readout | ./serialize | ./rampcheck". (Fungerar säkert dåligt om man inte har 100 sampel per kanal.)

Srupup config file samt nya register i SRUn:

- ☐ PSSCAN_VALUE bör vara 141 när vi kör 40 MHz DTC-klocka. Byter man plats på MCM-korten kan detta värde behöva ändras. Läs diagnostikregistren på adress 0x1028 och uppåt för att se om alla DTC-länkar fungerar. Fungerande register ska rapportera "0xBC50" som data på länken. Det bästa vore om hårdvaran själv kunde trimma mot mer än en kanal.
- ☐ SRU_CFG (register 0x20) does not correspond fully with the SRU documentation:
 - ☐ SRU_CFG[8] is SPILL_EN. Add to srupup and other programs that manipulate this register.
 - ☐ SRU_CFG[9] is the event counter selector. 0 selects internal event counter. 1 selects TLU event number.
 - ☐ SRU_CFG[24] is CHDVER (unused: TTC trigger message version selector?)
 - ☐ SRU_CFG[7] = BUSYINEN[1] is not used (busy inputs from neighbouring SRUs were removed).
 - ☐ SRU_CFG[6] = is repurposed as "EVENT_ACK_EN" (previously BUSYINEN[0]). A '1' in this bit will enable the "ALLOWED_EVENTS" register (register 0x46) which decrements for each event and activates BUSY when it reaches 0. Use this register to acknowledge singular or multiple events. A '0' in this bit ignores the "ALLOWED_EVENTS" register.
- ☐ New bits in SRUSTATUS:
 - ☐ SRUSTATUS[21] is '1' if word alignment is in progress. Should be '0' during normal operation.
 - ☐ SRUSTATUS[22] is '1' if ODELAY reference clock is functional (Used by DTC link output delays). Should be '1' during normal operation.

- ☐ **ALLOWED_EVENTS (register 0x46):**
Counts down each trigger. When 0, masks further triggers. Use to stall triggers from the SW side. Write 0x01h to allow a single trigger. Enable this function by writing a '1' in SRU_CFG[6].
Stalling one SRU will cause its BUSY/VETO flag to stay high after the event which also stalls the remaining SRUs. Thus, one register write to a single SRU is sufficient to control all SRUs, as long as other.
- ☐ After a TLU Event number has been caught by a partially accepted trigger, the SPTRIG_PERIOD register is used as a cooldown time during which additional trigger event numbers are ignored.
Set this time to a slightly longer time than the trigger latency (which can be the L0 latency or L1 latency depending on the settings of strig_config[2:1].
Without this cooldown, TLU triggers appearing in rapid succession can cause the wrong event number to be locked. (Events should not arrive in rapid succession if the TLU is properly configured).
- ☐ New register in SRU on address 0x25: DTC clock delay control.
Writes to this register will enforce a delay on the outgoing DTC clock (and thus the MCM/ADC sample clock) of the selected DTC channel in steps of 78 ps. Delays from 0 to ~2.4 ns can be accomplished. The delay value is relative to an internal reference clock (synchronized with the TLU clock if the TLU clock is selected).
 - Bit[21:16] = Channel number (values 0 to 39 are valid).
 - Bit[4:0] = delay value (values 0 to 31 are valid).
 In rare cases, delay changes may cause the DTC return link to malfunction. Read the corresponding data link registerj or start a DTC link error detection to detect this (register 0x45 DTCERRTEST and registers 0x10a0 - 0x10cf)).
If there is a problem, PSSCAN and/or WD_ALIGN may have to be performed after changing the delay values.
- ☐ New status/diagnostic registers in SRU:
Addresses 0x10d0 - 0x10f7 : DTC_OUTPUT_DELAY for each DTC channel
Current delay values for DTC link outputs (sample clock phase delays). 5-bit values: Each unit of the value implies a 78 ps delay vs the internal reference.

TLU:

- ☐ Jag tror att TLUn är inställd på att ignorera VETO-signalen från SRUn. Det spelar ingen större roll eftersom SRUn själv ignorerar inkommande triggers medan BUSY/VETO är aktiv, men det kan spela roll när vi använder fler SRUer.