Af DL1YDR, Andreas Franzen. Oversat og bearbejdet fra CQ DL 5-2016 af Peter Raabye, OZ5DW. Bringes med tilladelse fra DARC

Artiklen beskriver opbygning af et målesystem, baseret på Red Pitaya, til bestemmelse af gennemgangskurven for et krystalfilter. Det er et frit programmerbart open source-måleapparat, der kan erstatte flere andre apparater, fra signalgenerator til oscilloskop.

Red Pitaya er en single board-computer, baseret på en Xilinx Zync System on Chip (SoC). Den har to analoge indgange og to analoge udgange med 125 MHz sample rate og en opløsning på 14 bit. SoC'en kombinerer en FPGA med to ARM mikroprocessorkerner.

Tilslutninger

I stedet for harddisk bruges et micro SD-kort. Kommunikation med omgivelserne foregår enten via en seriel forbindelse over USB, eller over ethernet. Der er ikke mulighed for at tilslutte en skærm, så Red Pitaya forventes brugt sammen med en anden computer. Man kan f.eks. grundlæggende bruge Red Pitaya spectrum analyzer sammen med en WLAN-router og se resultaterne på en tablet pc i en webbrowser.

Muligheder for programmering

Der er flere muligheder, hvis man vil programmere sine egne målefunktioner. Man kan starte Red Pitaya over netværket fra en PC, og på PC'en køre et program, der styrer signalgeneratoren og måleopsamlingen. Man kan også programmere ARM-processorerne på Red Pitaya direkte, og beregne direkte på rådata på Red Pitaya. Her kan man vælge at udnytte standardinstallationens muligheder for kommunikation, eller skrive sin egen sourcekode, der styrer FPGA'en. Endelig kan man lægge nye logiske kredsløb i FPGA'en.

Programmering af FPGA'en

Her beskrives et enkelt eksempel på programmering af FPGA'en i Red Pitaya. I standardprogrammellet blinker den første gule LED i stikrækken med ca. 0,9 Hz, idet AD-clock'ens 125 MHz deles ned med 227 = 134217728.

Den binære tæller er programmeret i FPGA'en og forbindes med den relevante udgang. LED'en blinker også, når ARM-processorerne er sat i stå med en Halt-kommando.

Her vises, hvordan man indsætter en ekstra AND-gate med tre indgange. Denne kan forbindes med den tidligere beskrevne udgang fra den binære tæller og med de to næste, lavere udgange. Udgangen af AND-gaten fører til LED'en. Som følge af det, blinker denne med samme frekvens, men kun 1/8 af tiden.

Beskrivelse af programmeringsprocessoren

Denne bygger på Pavel Demins beskrivelse på hans GitHub repository "red-pitaya-notes" [1]. Der beskrives også udvikling af en SDR-enhed, baseret på Red Pitaya.

Den benyttede PC er baseret på en i5-4460 CPU og kører Ubuntu 14.04.2LTS i 64 bit. Kompilering af binærfilerne til FPGA'en tager ca. 10 minutter på denne PC. Der er installeret en del programmer ud over standardinstallationen.

Her følger Pavel Demins beskrivelse (bemærk, at indrykkede linier skal skrives i forlængelse af linien før):

sudo apt-get

-no-install-recommends

install build-essential git curl

ca-certificates sudo

libxrender1 libxtst6 libxi6

lib32ncurses5

gcc-arm-liniux-gnueabihf

g++-arm-linux-gnueabihf

bc u-boot-tools

libncurses5-dev

qemu-user-static

binfmt-support dosfstools

parted debootstrap

Derudover skal man installere gtkterm som seriel konsol og plotutils til graph-kommandoer med:

sudo apt-get install gtkterm

plotutils

og så skal man med:

sudo ln -s make /usr/bin/gmake

sikre at man kan kalde make som gmake.

Af hensyn til installationen oprettes brugeren redpit med home i /home/redpit. Man skal huske, at en bruger kun kan benytte sudo til at få midlertidige administratorrettigheder, hvis brugeren er knyttet til gruppen sudo.

For at kunne forbinde serieporten med Red Pitaya skal brugeren også tilhøre gruppen dialout:

sudo adduser redpit sudo

sudo adduser redpit dialout

Disse kommandoer benyttes med det brugernavn, man oprettede Ubuntuinstallationen med. Disse gruppetilhør er først aktive, når redpit logger på næste gang.

Menupunkterne findes i standardinstillingen i øvrigt ikke øverst på skærmen, men helt oppe i skærmranden, hvor de viser sig, når musen køres derop.

Programmeringsprogrammellet

Man skal hente Xilinx-Vivado_SDK_LIn_2015.2_0626_1.tar.gz fra producenten af Zync, Xilinx [2].

På deres hjemmeside skal man oprettes som bruger. Den relevante udgave er "Vivado Design Suite – 2015.2, Full Product Installation, Vivado 2015.2: Full Installer For Linux Single File, Download Including SDK". Filen er på 4,58 GB og var den aktuelle, da denne artikel blev skrevet.

Til installationen behøves skriverettigheder til /opt:

sudo chmod a+w /opt

Man opretter nu et bibliotek Xilinx, og skifter til dette:

mkdir /opt/Xilinx

cd /opt/Xilinx

Den hentede fil kopieres til dette bibliotek og udpakkes:

tar -xzf Xilinx_Vivado_SDK_LIn_2015.2_0626_1.tar.gz

Dette opretter et bibliotek, som vi skifter til:

cd Xilinx_Vivado_SDK_Lin_2015.2_0626_1

Pavel Demin anbefaler, at man i xsetup ændrer uname -i til uname -m. I den anvendte Ubuntu-version har de imidlertid samme effekt.

Nu starter man installationen med ./xsetup. Man vælger gratislicensen "Vivado WebPack. Man fravælger option "Devices 7 series, og tilvælger "software development kit".

Derefter slås "acquire or manage a license key" fra, og man sætter path og andre variable med:

source /opt/Xilinx/Vivado/2015.2/settings64.sh

så Vivadofilerne kan findes. Scriptet skal køre hver gang man logger på, og det kan gøres med den skjulte fil /home/redpit/.bashrc.

Vedrørende licensen

Der oprettes et trusted storage area til opbevaring af licensen. Man skifter bibliotek:

cd /opt/Xilinx/Vivado/2015.2/bin/unwrappet/lnx64.o

og kører

sudo ./install_fnp.sh

Dette opretter biblioteket /usr/local/share/macrovision/storage. Dernæst opretter man et bibliotek under home:

cd

mkdir Xilinx

cd Xilinx

mkdir Lizenz

og skifter til det:

cd Lizenz

Der opretter man en licensanmodning med:

xlicclientmgr -cr request.html

Man kan ikke åbne filen med Firefox. Man kan dog åbne den samtidigt oprettede fil request..html. Under create new licenses, vælger man "Activation based licenses, Vivado WebPack license Activation – no charge". Man kører "Activate Node-Locke License" og får en email. Ved hjælp af denne henter man filen Xilinx_license.xml, der kopieres til biblioteket Lizenz. Fra dette bibliotek starter man:

xlicclientmgr -p Xilinx_license.xml

Sourcekoden

Nu installeres sourcekoden fra Pavel Demins red-pitaya-notes. Man går på GitHub [1] og leder efter red-pitaya-notes. Med valget Dwoload ZIP henter man arkivet red-pitaya-notes-master.zip, og pakker det ud med:

unzip red-pitaya-notes-master.zip

Her findes under project/red_pitaya_0_92, den på skrivetidspunktet, seneste stabile version.

Oprettelse af AND-gaten

Verilog-sourcen til styring af den gule LED findes i filen red_pitaya_hk.v. Linie 124 indeholder ordren om at at binde udgang 26 fra binærtælleren led_cnt til LED 0:

led_reg[0] <= led_cnt[26];

For at indsætte AND-gaten, ændres linien til:

led_reg[0] <= led_cnt[26] &

led_cnt[25] & led_cnt[26];

Denne linie giver compileren besked på at sætte en AND-gate med tre indgange op, og at forbinde disse tre indgange med udgangene 26, 25 og 24 fra binærtælleren led_cnt, og forbinder udgangen med registeret led_reg[0].

Kompilering

I biblioteket red-pitaya-notes-master finder man en makefile, der sammenfatter de nødvendige kommandoer.For at kompilere går man i et terminalvindue til biblioteket og skriver:

make NAME=red_pitaya_0_92.bit

Når det er kørt findes underbiblioteket tmp til red-pitaya-notes-master, og deri den nye fil red_pitaya_0.92.bit, der har de binære data, der skal sendes til Zync FPGA'en.

Pavel Demin har et Pythonprogram, der laver en .bin-fil af .bit-filen, og derefter sender den til FPGA'en.

I Red Pitaya's dokumentation anbefales det at hente den gamle version, ISE, af udviklingsmiljøet, for at bruge programmet promgen til samme formål.

Pythonprogrammet er afprøvet og det eneste resultat var at man får meldingen Found swapped sync word, når man sender .bit-filen til FPGA'en, og Found normal sync word, når man sender .bin-filen. Man behøver således ikke at ændre .bit-fil til .bin-fil.

Den grafiske brugerflade

Ud over .bit-filen dannes en projektfil red_pitaya_0_92.xpr, der kan åbnes med Vivado's grafiske brugerflade. Vivado startes fra et terminalvindue med kommandoen vivado.

Programmering af FPGA'en

For at kunne programmere FPGA'en, må man finde dens IP-adresse. Det kan gøres i et terminalvindue med cat /dev/kmsg. Når man så kobler den serielle snitflade til via USB, kommer meldingen FTDI USB Serial Device converter now attached to ttyUSB0. Device'n kan nu kontaktes som /dev/ttyUSB0.

Man afslutter cat med ctrl-c og starter en seriel konsol på den fundne device:

gtkterm – p /dev/ttyUSB0 -s 115200

Når man nu tilslutter den yderste USB-forbindelse fra Red Pitaya til strømforsyningen, kan

man se startsekvensen på den serielle konsol. Der står der noget i retning af Lease of 192.168.2.39 obtained,…

Alternativt kan man starte ifconfig på Red Pitaya via den serielle konsol. Her vises i stedet f.eks. inet addr:192.168.2.39. Når man så kender IP-adressen 192.168.2.39 kan man kontakte Red Pitaya over netværket.

I et normalt terminalvindue på pc'en går man til biblioteket re-pitaya-noes-master. For at kopiere filen tmp/red_pitaya_0_92.bit til Red Pitaya, skriver man:

scp tmp/red_pitaya_0_92.bin

root@192.168.2.39:/root

Man blive bedt om et password, og skriver "root". På den serielle konsol kan man nu finde filen ved at skrive først cd /root, og bagefter ls.

Red Pitaya omprogrammeres nu med kommandoen:

cat red_pitaya_0_92.bin >/dev/xdevcfg

og den gule LED blinker nu med kun 12,5% duty cycle.

Giver man kommandoen reboot, blinker LED'en igen med 50% duty cycle.

C-programmet

Figur 1. Diagram af måleopstillingen.

Figur 1 viser diagrammet for krystalfiltret. Udgang 1 på Red Pitaya belastes med de anbefalede ca. 50Ω. Ved hjælp af de to 330Ω modstande ser krystalfiltret ca. 330Ω på både indgang og udgang.

Krystallet er den lave version HC49/U-S.

C-programmet, der alene bruges til at måle frekvenskarakteristikken for det viste krystalfilter, kan hentes på EDRs hjemmeside. Filen hedder Messung.c.

Der hentes det maksimale antal målepunkter, 16384. "cal" er indgangskalibrering med højre forstærkning til 2V over 16384 trin a 14 bit. f_adc er samplingfrekvensen og f0 er krystallets nominelle frekvens. Der måles 801 punkter fra -4 til +4 kHz, i forhold til f0. Der sendes en generate-kommando, der sætter et 1V S-S sinussignal på udgang 1 og venter 50 ms. Kommandoen startes igen og der køres en acquire-kommando, der måler de 16384 punkter uden kompression, svarende til en måletid på 131 µs. Kommandoen genstartes og output sendes til en pipe, hvorfra resultatet kan udlæses.

Målepunkterne læses derefter i det følgende loop og udsættes for en windowing-funktion, der har nul for begge ender. Vinduet stiger og falder jævnt, og har en middelværdi på 1. Windowingfunktionen giver undertrykkelse af randeffekterne af målevinduet, altså bla.a. en DC-undertrykkelse, også når måletiden ikke er et heltalligt antal gange målesignalets periode.

Målesignalet multipliceres herefter med et sinussignal på målefrekvensen, og resultatet summeres, tilsvarende med en cosinus. Til slut findes resultatet fra disse, og bliver logaritmeret.

Kald af C-programmet

C-programmet startes med shell script'et "Messung.sh", der ligeledes kan downloades fra EDRs hjemmeside:

bash Messung.sh

I første linie kaldes C-compileren ti ARM-arkitektur. Det betyder fremkomst af filen a.out. Denne kopieres til Red Pitaya og kan kaldes derfra, og output fra den omdirigeres til /tmp/Quartz.dat. Denne fil kopieres til pc'en og graph-kommandoen giver PostScript-filen tmp.ps. Den er ikke så pæn, så for at rette op på tegning af hjørner gøres følgende: med

streameditoren sed ændres alle setlinejoin med pop 1 selinejoin. Dermed sættes 1 for "afrundet" som parameter til setlinejoin. Tilsidst ændres filen til en .pdf (Figur 2).

Figur 2. Gennemgangskurve omkring nominel frekvens for filtret fra figur 1, målt med Red Pitaya.

Målingen over de 8kHz med 10Hz trin tog 134s.

Public Key Authentication

Når man kalder scp på Red Pitaya, skal man hver gang identificere sig som root. Dette kan man forhindre med Public Key Authentication. Man skal som bruger redpit skrive:

ssh-keygen -t rsa -b 4096

Det bekræftes med tre gange Enter, man lader altså password være tomt. Der fremkommer nu en offentlig nøgle, som man overfører til Red Pitaya hver gang man nystarter den:

ssh-copy-id -i ~/.ssh/id_rsa.pub

root@192.168.2.39

Derefter spørger Messung.sh ikke om password.

Kilder

[1] Red Pitaya: http://repitaya.com

[2] GitHub: https://github.com

[3] Xilinx: www.xilinx.com

Figur 3. Red Pitaya er en interessant blanding af computer og måleapparat, og er anvendelig til en del målinger i amatørradio.

Figur 4. Red Pitaya'en i gang med at måle på krystalfiltret.