

Övningsuppgifter

till
Föreläsning
i

TSEA52 DIGITALTEKNIK I

Konstruktion av mindre digitala system med CPLD

Version: 1.2

2018 (OVA, MK)

1. Inledning

Syftet med föreläsningarna är att presentera tankar och idéer till hur man konstruerar mindre digitala system med hjälp av det härvarubeskrivande språket VHDL. VHDL är ursprungligen skapat för att kunna simulera komplex hårdvara innan tillverkning sker. Därefter har man insett att en delmängd av språket är syntetiserbar, varför det idag finns rätt kraftfulla syntesverktyg som förstår VHDL.

Målet är att deltagarna efter föreläsningarna ska:

- Känna till och förstå funktionen på ett antal digitaltekniska standardblock
- På ett strukturerat och hårdvarunära sätt kunna lösa digitaltekniska uppgifter genom att designa blockscheman med standardblock
- Kunna översätta blockscheman till VHDL enligt god kodstandard

Samtliga uppgifter är tillräckligt små för att få plats i en CPLD av typen XC9572, och som är den krets som kommer att användas på lab 4 i kursen. Uppgifterna är också så pass små så att hierarkiska strukturer bör undvikas. Användningen av ”components” kommer att läras ut i senare kurser där behovet är mer påtagligt. **Man bör också undvika alltför stora processatser**, dvs en processats/block brukar vara lagom. En lämplig lösningsgång är att först rita ett blockschema och som därefter översätts till VHDL. Lämpliga typer av block: Räknare, Register, Komparatorer, Multiplexrar, PROM, Vippor och grindar, Kombinatoriska funktioner samt Sekvenskretsar med tillhörande tillståndsdigram. Exempel på digitaltekniska byggblock finns i ett separat dokument på kurshemsidan.

Saker att tänka på är bland annat:

- Namngivning av signaler
- Lämpliga datatyper (`std_logic`, `std_logic_vector`, `unsigned`)
- Konstruera ett synkront system, dvs alla vippor ska ha samma klocksignal
- Synkronisera de insignaler som påverkar synkrona ingångar.

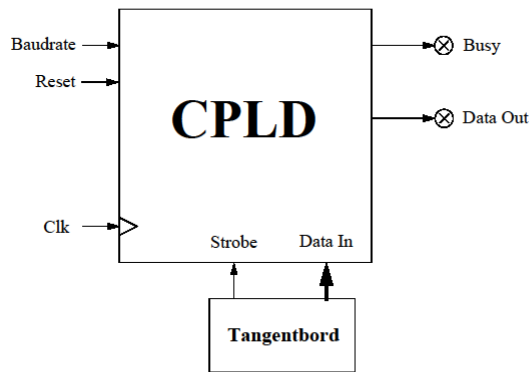
Systemklockan bör vara relativt hög, till exempel 8 MHz. Detta för att någon enstaka klockpuls fördröjning inte ska påverka slutresultatet, utan snarare tillåta stabilare och mer lättbegripliga lösningar.

Uppgifterna kan vanligtvis lösas på flera olika sätt, och föreläsningen gör inte anspråk på att hitta de smartaste varianterna, utan fokus ligger snarare på att presentera bra översättningar till VHDL.

Efter föreläsningarna kommer bland annat lösningsförslag och exempelkod att läggas upp på kurshemsidan.

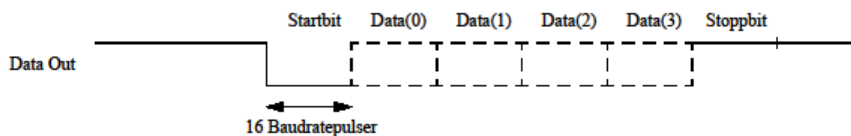
2. Övningsuppgifter

Uppgift 30: Skapa ett digitalt system som får information om fyra bitar från ett tangentbord, och som därefter sänder ut detta på seriell form. Formatet ska vara Startbit, databitar samt en stoppbit, se figur längre ner på sidan. Medan sändningen pågår ska detta indikeras med en LED (Busy). Information från tangentbordet ska under pågående sändning ignoreras.



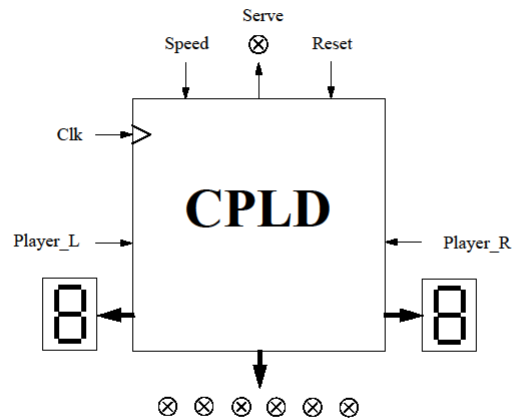
Signalen Baudrate bestämmer hastigheten på den utsända bitströmmen, och det är vanligt att den här signalen har en frekvens på 16x den utsända hastigheten. Tangentbordet ger ifrån sig en Strobe när det finns data att hämta. Strobe ligger hög så länge som en tangent är nertryckt. Data om senast nedtryckta tangent återfinns på Data In tills nästa tangent trycks ner. Signalen Reset ska asynkront avbryta eventuell sändning och återställa systemet i ett starttillstånd där det är redo att ta emot nästa knapptryckning.

Konstruera det digitala systemet med hjälp av VHDL. Rita först ett detaljerat blockschema. Glöm inte att synkronisera insignaler.



Format för asynkron seriell kommunikation med fyra databitar

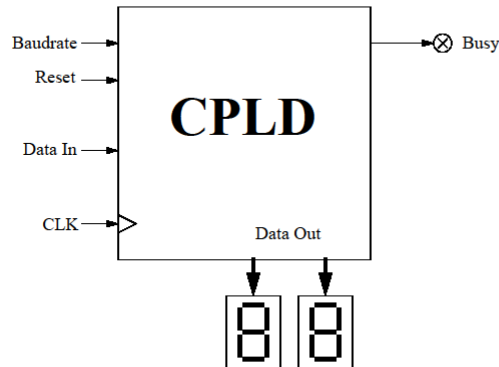
Uppgift 31: Skapa ett digitalt system som realiserar ett PING-PONG spel. En "Boll", markerad med en tänd lysdiod, ska förflyttas fram och tillbaka över en spelplan bestående av en rad lysdioder. Spelet fortgår så länge de bägge spelarna trycker ned sin knapp exakt då bollen befinner sig i respektive ändläge.



Den som vinner en boll får serva. Serve markeras med stillastående boll på den servandes sida, samt med en LED (Serve). Bollen börjar att röra sig då knappen trycks ned. Spelare V börjar att serva vid spelets start. Om en spelare trycker antingen för tidigt eller för sent vinner den andre bollen och får poäng. Likaså om bollen går ut och spelaren inte trycker, då vinner den andre bollen. Nytt serveläge ska komma automatiskt när en spelare har fått poäng. Det är bara den spelare som är i tur att trycka som tillåts att göra det, och om den andre spelaren likväl trycker utom tur ska detta ignoreras. Insignalen Speed ska bestämma hur fort bollen rör sig över spelplanen. Reset ska vara en synkron funktion som gör systemet redo för en ny match.

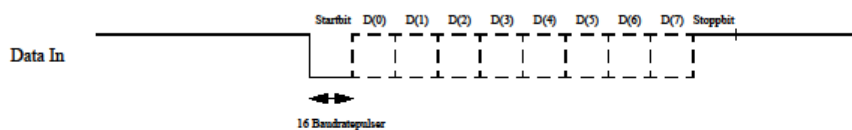
Konstruera det digitala systemet med hjälp av VHDL. Rita först ett detaljerat blockschema. Glöm inte att synkronisera insignaler.

Uppgift 32: Konstruera en mottagare för en seriell signal som innehåller: startbit, databyte (8 bitar) samt en stoppbit, se figur längre ner på sidan. En LED (Busy) ska lysa så länge som mottagning av en databyte inklusive start och stoppbit pågår. Resultatet ska dyka upp på sju-segmentsdisplyerna först när hela mottagningen är klar.



Avläsning av informationsbitarna sker normalt sätt mitt på för att få så stor marginal som möjligt. Det är också vanligt att man filtrerar bort falska startbitar genom att man kontrollerar att startbiten fortfarande är noll halvvägs in i startbiten. Reset ska synkront avbryta mottagning av

Konstruera det digitala systemet med hjälp av VHDL. Rita först ett detaljerat blockschema. Glöm inte att synkronisera insignaler.



Format för asynkron seriell kommunikation med åtta databitar