

Laboration i digital konstruktion I: Lab 4

Bengt Oelmann, Mitthögskolan, ITE

1.0 Sekventiell addera-och-skifta multiplikator

1.1 Syfte

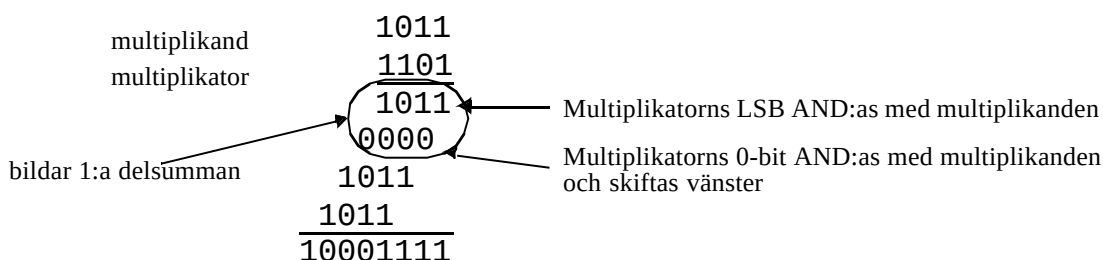
Syftet med den här laborationen är att ge insikt i hur man konstruerar en funktion med både en beräkningsenhet och en kontrollenhet.

1.2 Bakgrund

I de flesta typer av konstruktioner betraktas multiplikation som en komplex operation som både tar lång tid att utföra och kräver mycket logik för att implementera. En multiplikation kan implementeras som ren kombinatorik. För att reducera komplexiteten kan man dela upp multiplikationen i flera klockcykler som resulterar i enklare logik men multiplikationen görs istället under ett antal klockcykler. För att kontrollera vad som ska ske under de olika klockcyklerna måste beräkningsenheterna (datavägarna, eng. *data paths*) styras av en tillståndsmaskin (FSM).

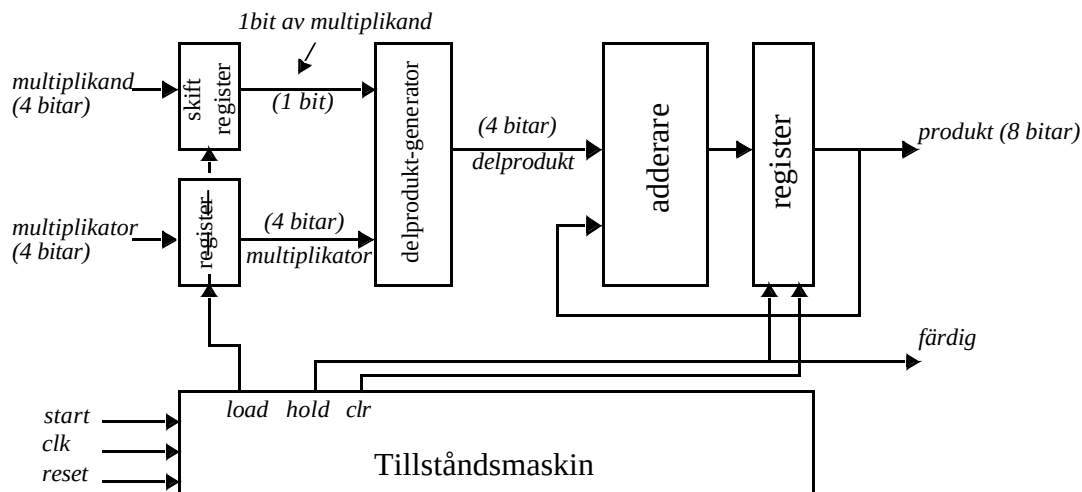
I den här laborationen ska en addera-och-skifta multiplikator konstrueras. Multiplikationen går till på samma sätt som i grundskolan: multiplikanden multipliceras med en siffra (eller bit) av multiplikatorn, och för varje mer signifikant bit av multiplikatorn skiftas delprodukten vänster. Exempel:

Multiplitera $1011 \cdot 1101 = B_{16} \cdot D_{16} = 8F_{16}$



Efter det att en delprodukt har bildats kan den adderas till den tidigare summan av delprodukter.

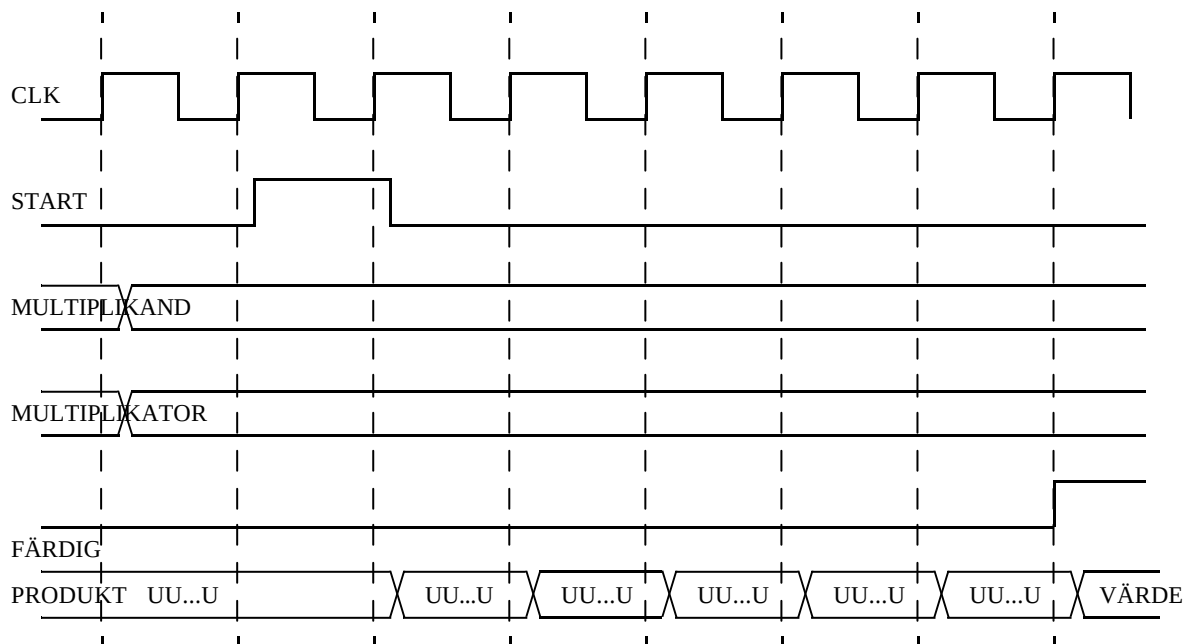
En addera-och-skifta multiplikator har följande struktur:



Tabell 1: kontrolls signaler

Signal	Funktion
clk	klocksignal som går till samtliga register (insignal)
reset	initiering av tillståndsmaskinen (insignal)
start	starta multiplikationen (intern signal)
clr	nollställ registret som innehåller summan av del-produkterna (intern signal)
load	ladda multiplikand och multiplikator (intern signal)
hold	bibehåll värdet på produkten (intern signal och utgång). Indikerar då multiplikationen är klar.

Timingdiagram:



1.3 Förberedelser

```
cd ~/VHDL
mkdir lab4
cd lab4
qhlib work
```

1.4 Utförande

Konstruera en skifta-och-addera multiplikator som kan multiplicera två positiva 4-bitars tal som resulterar i en produkt på 8-bitar. Multiplikationen ska initieras med en start-signal och då multiplikationen är färdig ska detta indikeras med en färdig-signal. Produkten ska ligga kvar tills nästa multiplikation har startats. Konstruktionen ska skrivas i syntetiserbar VHDL-kod. Multiplikatorn kan delas upp i flera komponenter som kopplas samman strukturellt.

1.5 Redovisning

VHDL-koden ska lämnas in till labhandledaren tillsammans med en beskrivning av in- och utsignaler (gärna som kommentarer i koden).