

G1
Maskinarkitektur
Efterår 2011

Jens Fredskov
Naja Mottelson (vsj465)
Søren Pilgård (vpb984)

17. september 2011

Indhold

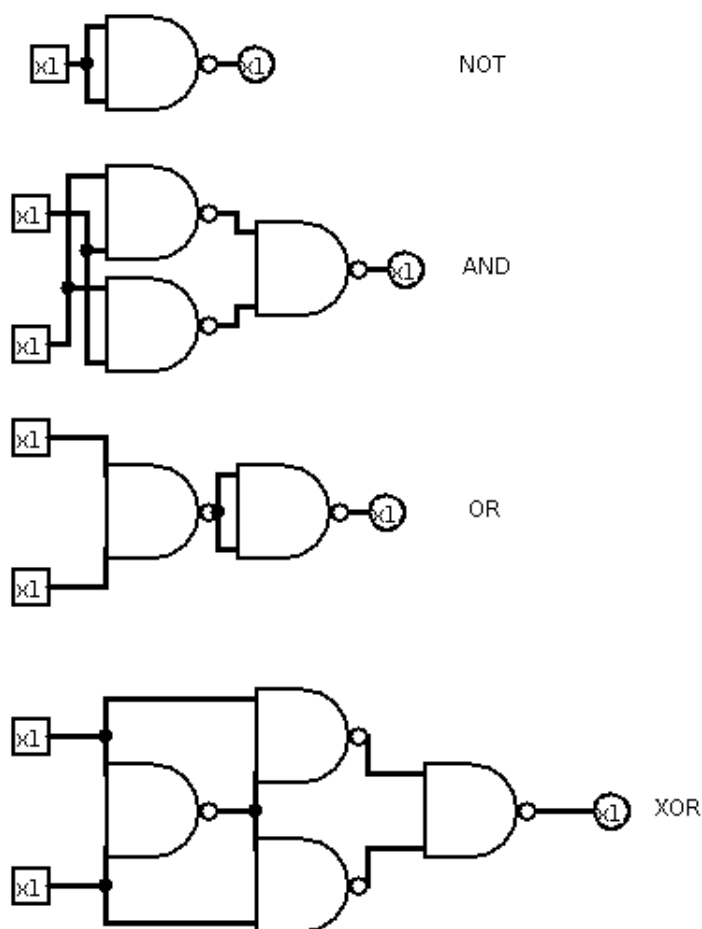
1	Indledning	3
2	g1-1	3
3	g1-2	4
3.1	Multiplexer	4
3.2	Adder	4
3.3	Overløbskontrol	4
4	g1-3	4

1 Indledning

Nærværende rapport tjener som dokumentation af gruppens arbejde med første godkendelsesopgave. Den indeholder korrekte og afprøvede løsninger på samtlige underopgaver.

2 g1-1

På Figur 1 ses vores implementation af de grundlæggende logiske gates (AND, OR, NOT, XOR) vha. NAND-gates. Som det ses følger vores implementering metoden i lærebogens Appendix C.



Figur 1: NOT-, AND-, OR- og XOR-gates af NAND-gates

3 g1-2

Overordnet er vores 4-bit ALU (se Figur 2) implementeret som en serie af fire 1-bit ALU'er (se Figur 3) efter samme logik som den 32-bit ALU der præsenteres i Appenix C-29. Her forbindes CarryOut-outputtet fra de mindre betydende bits til CarryIn-inputtet for de mere betydende. Nedenfor er en gennemgang af de operationer 1-bit ALU'en understøtter og deres implementering:

AND, OR ALU'ens grundlæggende logiske funktioner benytter de indbyggede gates i logisim.

NOR Outputtet til NOR udregnes som NOT A AND NOT B. Til dette benyttes multiplekserne Ainvert og Binvert.

Addér ALU'en benytter et Adder-modul (se Figur ??), som vi i gruppen har implementeret vha. logisims Combinatorial Analysis-værktøj og sandhedstabellen i Figur C.53.

Subtrahér Substraktionsfunktionen benytter samme Adder, blot med én af operanderne inverteret.

Set Less Than Set Less Than-operationen (SLT) er en komposit operation: Først sammenlignes A og B, hvilket giver resultatet 1 hvis $A < B$, 0 ellers. Herefter sættes alle inputtets bits til 0, med undtagelse af den mindst betydende bit som sættes til resultatet af sammenligningen. I forbindelse med 1-bit ALU'en giver SLT derfor ikke så megen mening (NB! Er dette overhovedet korrekt? Logikken virker da udmærket, hvis man tager Set som output. Men det er vel defineret således at Result er det eneste output vi interesserer os for?)

ZERO Zero-outputtet er separat fra Result-outputtet og beregnet ved at føre output-signalet fra de andre operationer igennem en XNOR-gate. (NB! Er Zero overhovedet defineret for 1-bit ALU'en?)

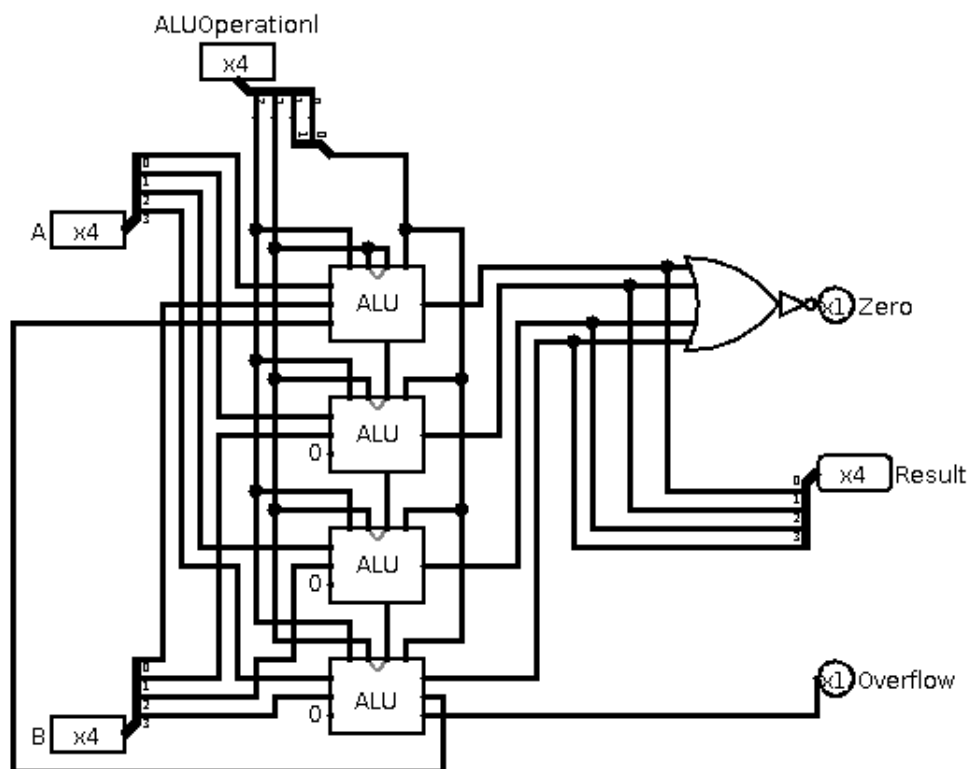
Som det ses adskiller seriens sidste 1-bit ALU (som håndterer den mest betydende bit i inputtet) sig fra de foregående ved at understøtte yderligere funktionalitet til håndtering af SLT-operationen samt at undersøge for overløb. Udover overløbsmodulet (se Figur 7) har vi implementeret denne del af logikken efter samme algoritme som beskrives i Appendix C-31-C-35.

3.1 Multiplexer

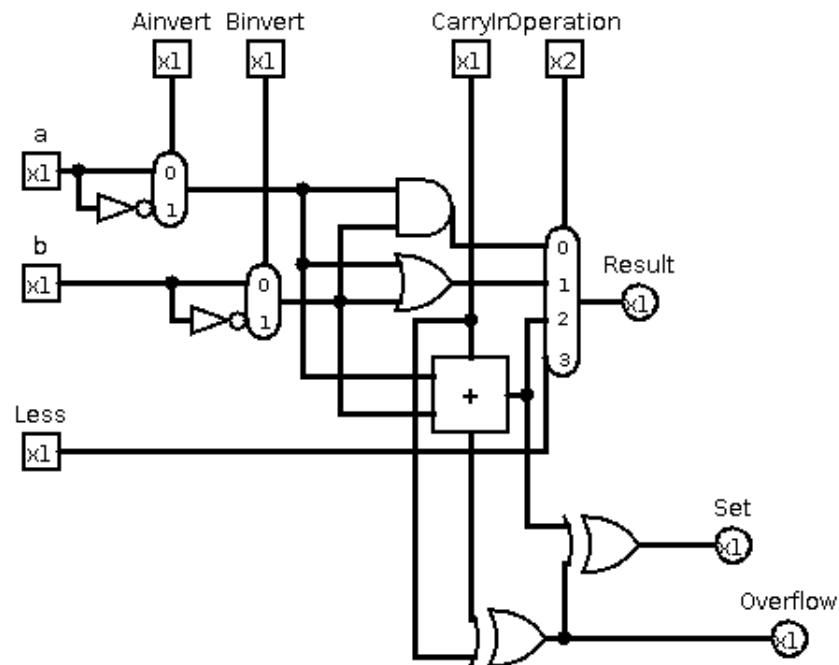
3.2 Adder

3.3 Overløbskontrol

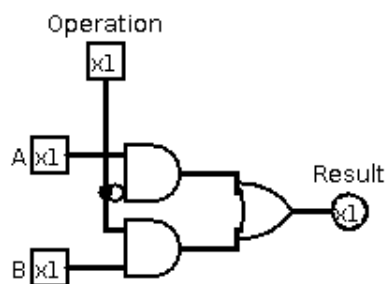
4 g1-3



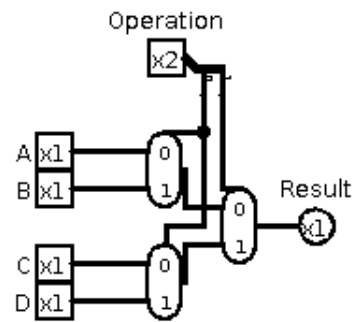
Figur 2: ALU, 4-bit



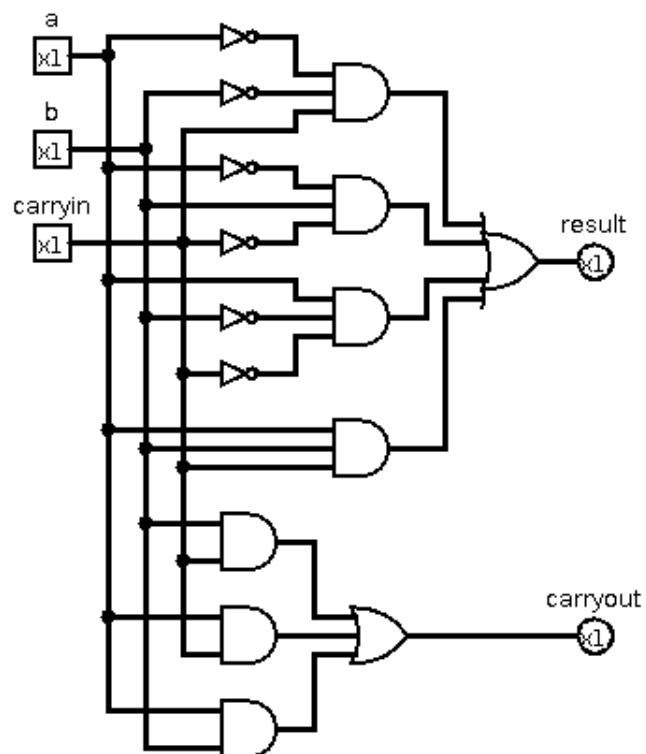
Figur 3: 1-bit ALU m. undersøgelse af overløb



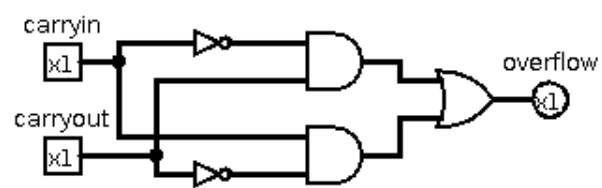
Figur 4: MUX, 2 bit IN



Figur 5: MUX, 4 bit IN



Figur 6: 1-bit adder



Figur 7: Logikken til undersøgelse af overløb