11:22



Lukas Ingold 20-123-998

GTI HS 20 Serie 10

Michael Baur, Tatjana Meier, Sophie Pfister

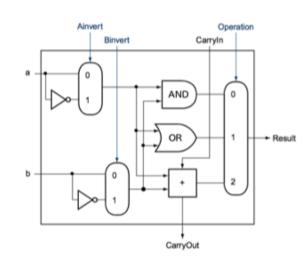
Die 10. Serie ist bis Montag, den 7. Dezember um 12:00 Uhr zu lösen und als PDF-Dokument via ILIAS abzugeben. Für Fragen steht im ILIAS jederzeit ein Forum zur Verfügung. Zu jeder Frage wird, falls nicht anders deklariert, der Lösungsweg erwartet. Lösungen ohne Lösungsweg werden nicht akzeptiert. Allfällige unlösbare Probleme sind uns so früh wie möglich mitzuteilen, wir werden gerne helfen. Viel Spass!

1 Operationen einer 1-Bit ALU (4 Punkte)

Für die gegebene 1-Bit ALU (siehe Abbildung) und die Funktion f, welche Werte müssen die Inputs Ainvert, Binvert, CarryIn und Operation haben, damit f(a,b) berechnet wird?

- (a) (2 Punkte) f(a,b) = (a NAND b)
- (b) (2 Punkte) f(a,b) = (a XOR b)

 $Tipp~zu~(a)\!{:}~({\bf a}~{\rm NAND~b}) = \neg (a \wedge b) \stackrel{{\rm de}~{\rm Morgan}}{=}$



2 Overflow bei der Addition (2 Punkte)

Ein Überlauf (Overflow) entsteht, wenn das Ergebnis einer Operation nicht mit der verfügbaren Hardware dargestellt werden kann.

- (a) (1 Punkt) Welcher Wertebereich (ganze Zahlen) kann durch ein 16-Bit Wort dargestellt werden, wenn die Zahlen
 - (i) im Zweierkomplement
 - (ii) unsigned (ohne Vorzeichen)

dargestellt werden?

(b) (1 Punkt) Ein n-Bit Addierer nimmt als Inputs zwei signierte n-Bit Zahlen (signiert = mit Vorzeichen, Darstellung im Zweierkomplement). Der Output hat auch die Länge n – ein allenfalls entstehender Übertrag aus dem Addierer kann wegen der fixen Wortlänge nicht vom Prozessor verarbeitet werden.

Beschreibe, wie man (ohne das Übertrags-Bit zu benutzen) erkennen kann, dass eine Addition zu einem Overflow geführt hat. Begründe deine Antwort.

3 4-Bit ALU (4 Punkte)

Abbildung). Diese 4-Bit ALU nimmt als Input eine 16-Bit lange Instruktion, welche wie folgt aufgebaut ist:

Eine 4-Bit ALU kann realisiert werden, indem man vier 1-Bit ALUs zusammenschaltet (siehe

unused a0 a1 a2 a3 b0 b1 b2 b3 Alayert Blavert Carryla Operation

(a) (1 Punkt) Welche Boolesche Funktion wird mit folgender Instruktion berechnet?

(b) (3 Punkte) Bestimme die Werte von Resulto, R

Overflow nachdem die Instruktion aus (a) fertig berechnet wurde.

Binvert

- 1. Tipp zu (b): Der CarryOut jeder 1-Bit ALU wird immer berechnet, egal welchen Wert Operation annimmt.
- Tipp zu (b): Die unterste 1-Bit ALU hat eine Overflow-detection (siehe Vorlesung).
 Tipp zu (b): Die MSBs (Most Significant Bits) sind a3, b3 und Result3. Die LSBs (Least Significant Bits) sind a0, b0 und Result0.

CarryIn

Operation

Overflow

=b a+b=r

a0 — CarryIn b0 — ALU0
Less
CarryOut

a1 — CarryIn b1 — ALU1
0 — Less
CarryOut

a2 — CarryIn b2 — ALU2
0 — Less

CarryIn

ALU3

Less

b3

ainvert =
$$\Lambda$$
b: α

1.) a) f(a,b) = a NAND $b = 7(a \wedge b)$

Operation = 1 => DE MORGAN:
$$7(a \wedge b) = 7a \vee 7b$$

Carry $In = D$

b) $f(a,b) = a \times 0R \ b = a \oplus b = 7ab + a7b$

$$Ainvert = 0$$
 abr
 $binvert = 0$ 0 0

0 1

geht un
$$-32^{17}68$$
 bis $+32^{17}67$

ii) geht un 0 bis $65^{17}635$

2.) a)

OPERATION = 2

Vorzeichen wo keiner sein sollte zwei Positive Zahren addert schen keine Negative Zahl V

Result

(Calch - In)

Overflow = 1

 $Set = 0 \qquad (1>0)$

Oveflow = 1