EE 3755, Fall 2002

Solutions of HW # 5

EE 3755, [redacted] ①.

Solutions of Homework# [redacted] 5

①.

$30 \rightarrow (V/IR[6:15] \wedge (\overline{nff \oplus vff}, cff, \overline{cff}, zff, \overline{zff}, nff, \overline{nff}, vff, \overline{vff}, nff \oplus vff))$

/(1).

Since the overall branch function cannot be complemented, $\overline{zff} \vee (nff \oplus vff)$ cannot be tested and therefore BGT can no longer be realized.

②. The function of interest is $\overline{zff} \wedge cff$. Observe that $\overline{zff} \wedge cff = \overline{\overline{zff} \wedge cff} = \overline{zff \vee \overline{cff}}$. Therefore, according to the details provided on page 2 of the handout entitled "First example computer: Part ③" the field IR[5:15] must be

IR[5:15] = 000 11 000000

- 000: regular branch.
- 11: $\overline{C}$ Z
- 000: branch on complement function being true.

[3] In the description below, DCD is a 4-to-16 decoder.

52 BBUS = (R0 ! R1 ! .... ! R15) * DCD(IR[12:15]);
ABUS[0:27] = (28'T0 ! $\overline{28'T0}$) * ($\overline{IR[12]}$, IR[12];
ABUS[28:31] = IR[12:15];
OBUS = (ABUS ! BBUS) * (IR[10], $\overline{IR[10]}$);
MD ← OBUS.

53 ABUS = same as before;
BBUS = (R0 ! R1 ! .... ! R15) * DCD(IR[6:9]);
cin = same as before;
OBUS = same as before;
(R0 ! R1 ! .... ! R15) * (($\overline{IR[2] \wedge IR[3]}$) ∧ DCD(IR[6:9])) ← OBUS;

cff * (same as before) ← same as before;
zff ← same as before;
nff ← same as before;
[illegible]
vff * (same as before) ← same as before;
→(100).

(4) In the description below, DCD is a 3-to-8 decoder.

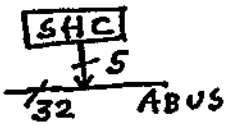
60 ABUS = 27TO, IR[11:15]; BBUS = $\overline{32TO}$; cin = 0;
OBUS = ADD[1:32](ABUS; BBUS; cin);
SHC ← OBUS[27:31].

61 BBUS = (R0 ! R1 ! ... ! R7) * DCD(IR[7:9]);
OBUS[0] = (BBUS[0] ! BBUS[1] ! BBUS[31] ! cff ! 0) *
($\overline{IR[5]} \wedge \overline{IR[6]} \wedge IR[10]$, $\overline{IR[10]}$, $IR[5] \wedge IR[6] \wedge IR[10]$,
$IR[5] \wedge \overline{IR[6]} \wedge IR[10]$, $\overline{IR[5]} \wedge IR[6] \wedge IR[10]$);
OBUS[1:30] = (BBUS[0:29] ! BBUS[2:31]) *
($IR[10]$, $\overline{IR[10]}$);
OBUS[31] = (BBUS[30] ! BBUS[0] ! cff ! 0) *
($IR[10]$, $IR[5] \wedge IR[6] \wedge \overline{IR[10]}$,
$IR[5] \wedge \overline{IR[6]} \wedge \overline{IR[10]}$, $\overline{IR[5]} \wedge \overline{IR[10]}$);
(R0 ! R1 ! ... ! R7) * DCD(IR[7:9]) ← OBUS;
cff * $(\overline{IR[5] \wedge IR[6]})$ ← (BBUS[31] ! BBUS[0]) *
($IR[10]$, $\overline{IR[10]}$).

62 ABUS = 27TO, SHC; BBUS = $\overline{32TO}$; cin = 0;
OBUS = ADD[1:32](ABUS; BBUS; cin);
SHC ← OBUS[27:31];
→ (V/SHC, $\overline{V/SHC}$) / (61, 100).

Some explanations follow on next page.

Explanations: The reason why I used a separate step 62 to be responsible for decrementing and testing SHC is because I assumed that the decrementation of SHC is performed by the adder of the ALU and involves the buses ABUS BBUS and OBUS. I assumed that the shift counter SHC is a connection to ABUS



If a dedicated decrementer was to be used for decrementing SHC in a way that the buses (ABUS BBUS OBUS) were not to be involved, then steps 61 and 62 could have been combined in one step.

5 There are hardware benefits obtained. Look at step 22 for example; (page 9 of handout "first example computer: Part (2)").

Due to the fact that SBC and SUB have adjacent opcodes as well as SUB and CMP have adjacent opcodes, the condition for connecting $\overline{MD}$ to ABUS, for example, is the simplified expression

$\overline{IR[0]} \wedge \overline{IR[1]} \wedge (\overline{IR[2]} \vee IR[3])$

The hardware implementation of the above expression relies on one OR and one AND gate. If opcodes were not adjacent, the above condition would have been more complex requiring more hardware.

[6] Again hardware benefits are obtained. Look how similar steps 22 and 53 are. Most of the connecting and clocking conditions are the same between steps 22 and 53.