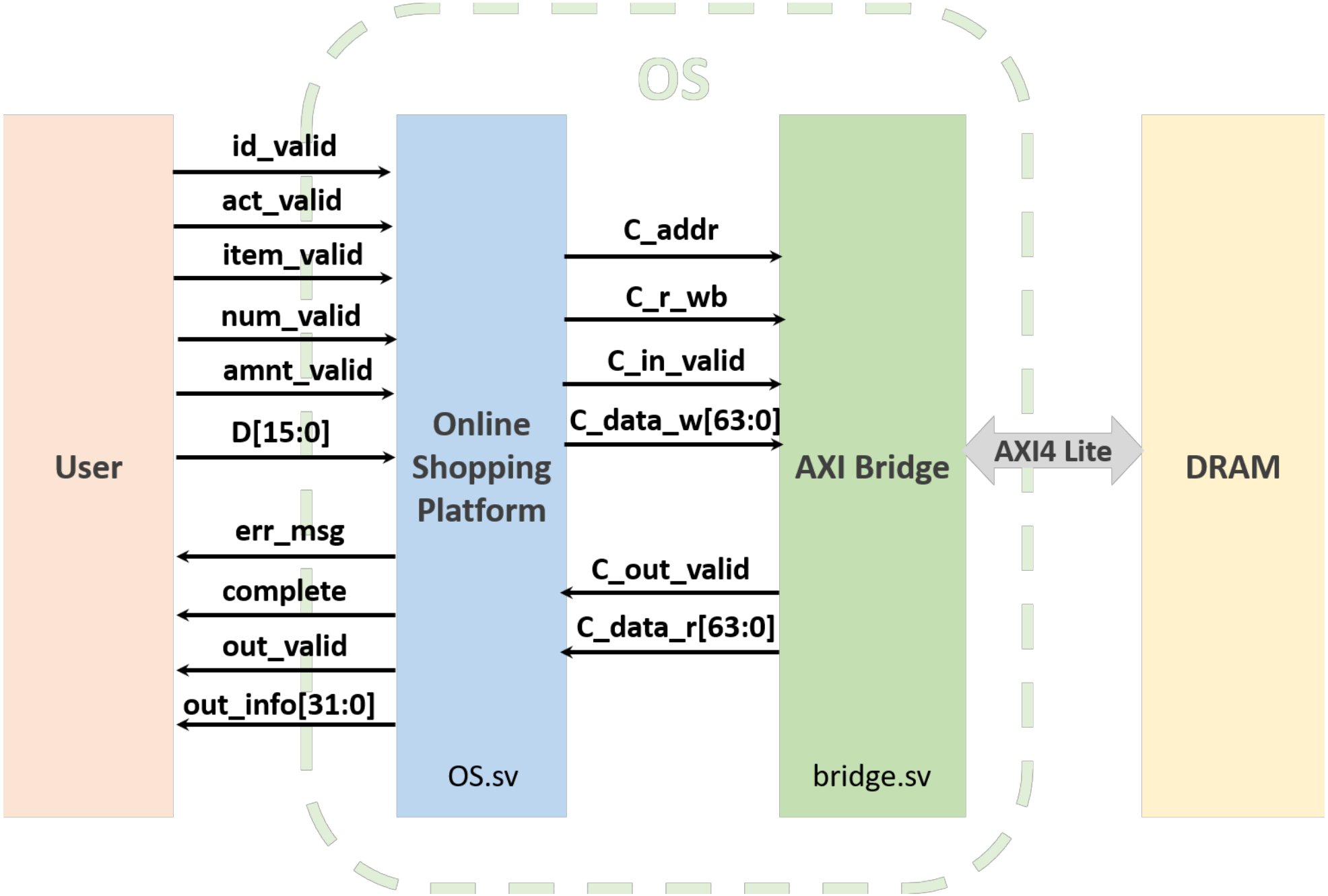
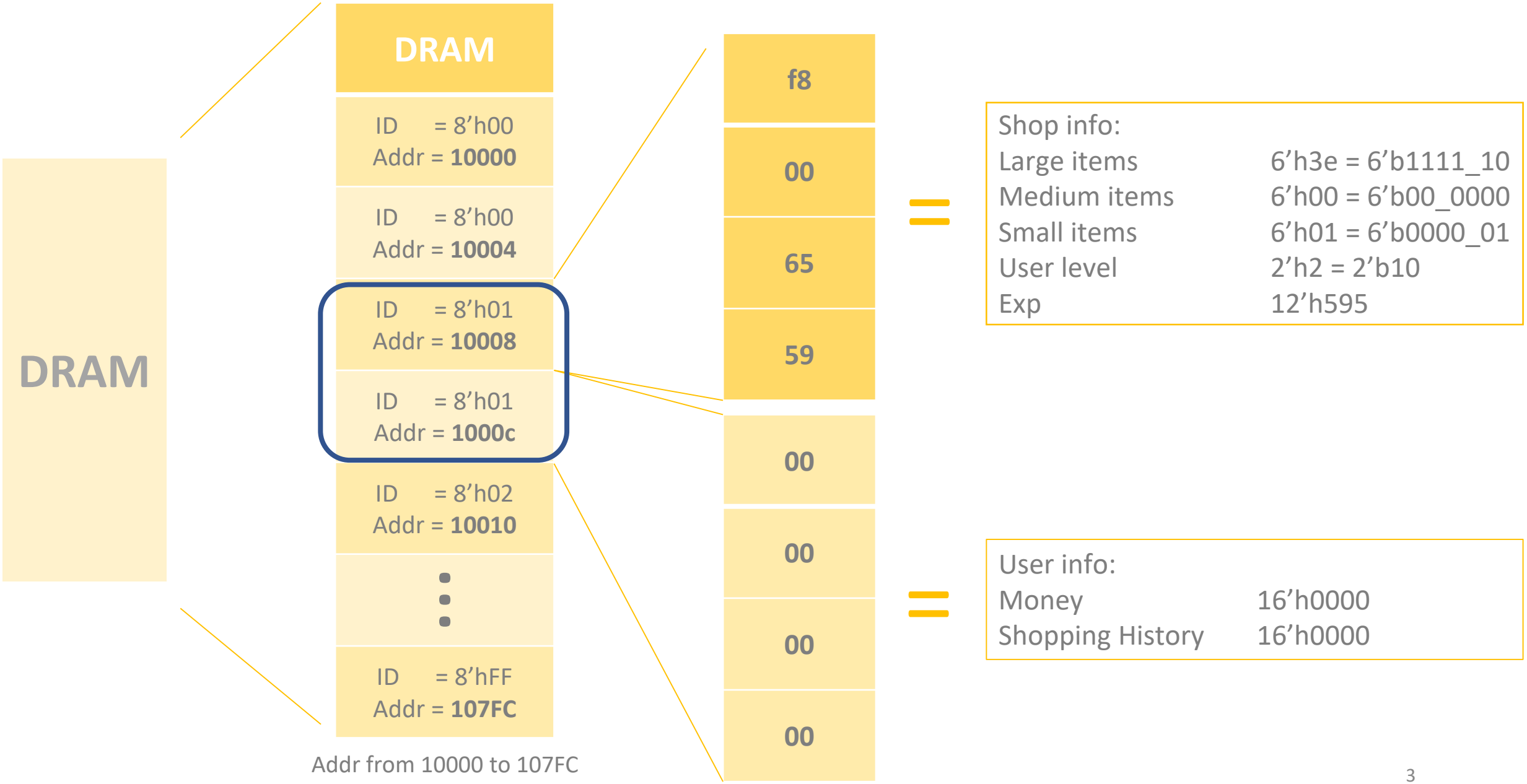
Lab09 Exercise note


OS
User
id_valid
act_valid
item_valid
num_valid
amnt_valid
D[15:0]
Online Shopping Platform
OS.sv
C_addr
C_r_wb
C_in_valid
C_data_w[63:0]
err_msg
complete
out_valid
out_info[31:0]
C_out_valid
C_data_r[63:0]
AXI Bridge
bridge.sv
AXI4 Lite
DRAM

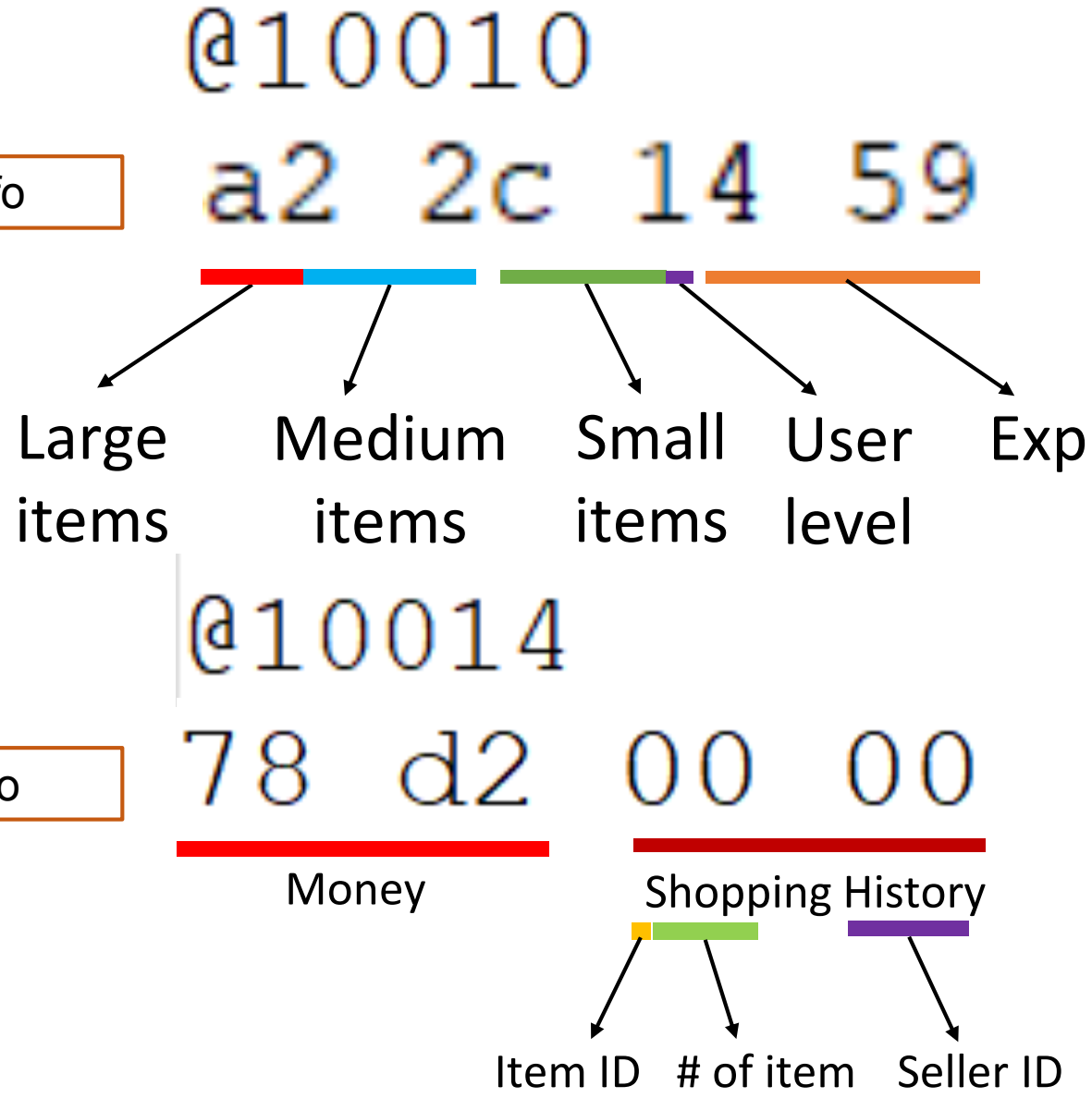

DRAM
DRAM
ID = 8'h00
Addr = 10000
ID = 8'h00
Addr = 10004
ID = 8'h01
Addr = 10008
ID = 8'h01
Addr = 1000c
ID = 8'h02
Addr = 10010
ID = 8'hFF
Addr = 107FC
Addr from 10000 to 107FC
f8
00
65
59
00
00
00
00
=
=
Shop info:
Large items 6'h3e = 6'b1111_10
Medium items 6'h00 = 6'b00_0000
Small items 6'h01 = 6'b0000_01
User level 2'h2 = 2'b10
Exp 12'h595
User info:
Money 16'h0000
Shopping History 16'h0000

.dat file example

```
@10000
e1 32 1c 5e
@10004
00 00 00 00
@10008
f8 00 65 59
@1000c
00 00 00 00
@10010
a2 2c 14 59
@10014
78 d2 00 00
@10018
f7 59 57 42
@1001c
00 00 00 00
@10020
f1 50 17 63
```

Shop info

User info



Note: The data format in dram is different from the format of out_info. Make sure you have dealt with this difference.

DRAM note

- You may modify the following part in ../00_TESTBED/pseudo_DRAM.sv.

DRAM latency →

```
parameter DRAM_R_latency = 1;
parameter DRAM_W_latency = 1;
parameter DRAM_B_latency = 1;
```

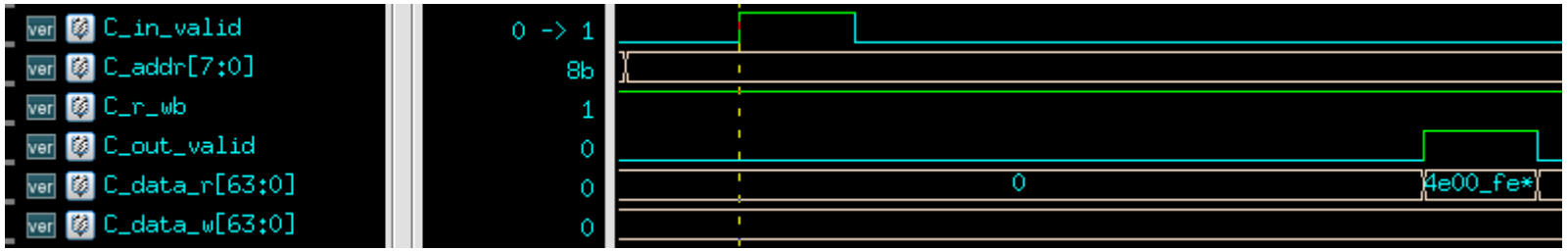
- If you want to initialize dram in pattern, you may use the following code.

Declaration of dram reg array →

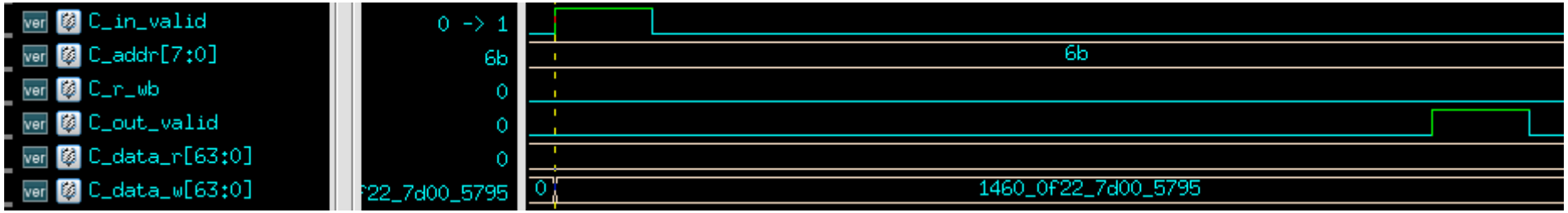
```
parameter DRAM_p_r = "../00_TESTBED/DRAM/dram.dat"

logic [7:0] golden_DRAM[ ((65536+256*8)-1) : (65536+0)] ;

initial $readmemh(DRAM_p_r, golden_DRAM);
```

Bridge

When C_in_valid is high, bridge will check C_r_wb

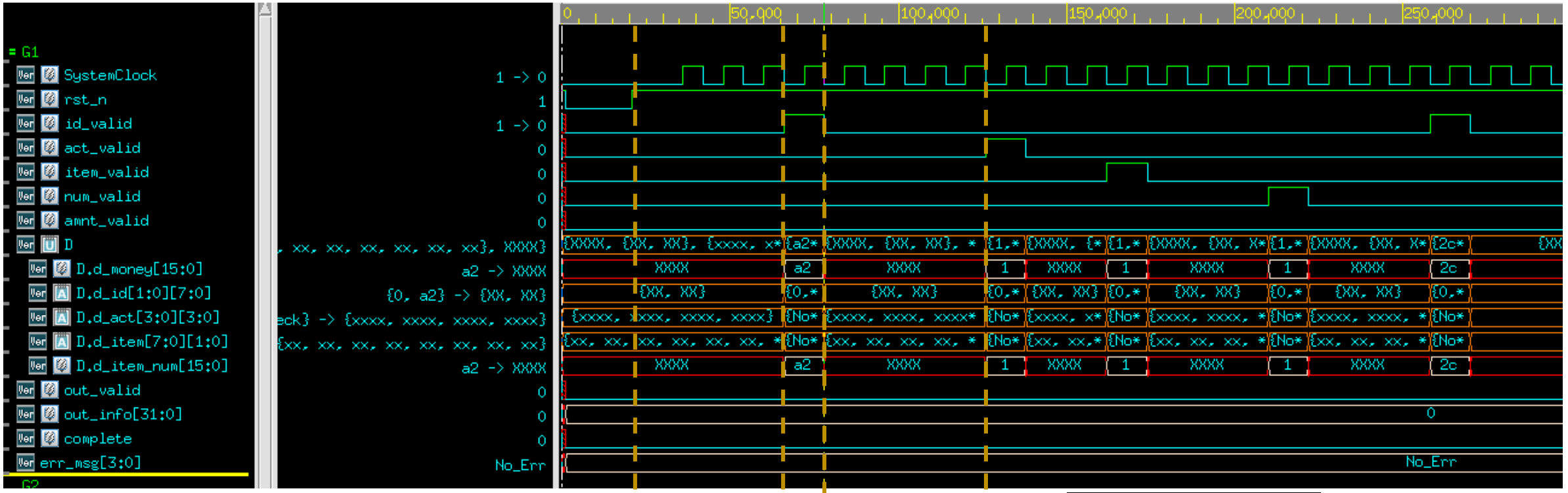


If C_r_wb is 1 (read), then it will base on C_addr to find the corresponding address in dram. When the data from dram is valid, it will pull high C_out_valid and return the value from dram.



If C_r_wb is 0 (write), then it will base on C_addr to find the corresponding address in dram. And then it will write C_data_w to that address. After writing, it will pull high C_out_valid to indicate that the write process is done.

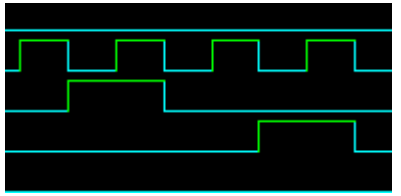
Start of the system



Delay is not fixed

Gap is not fixed

1 cycle

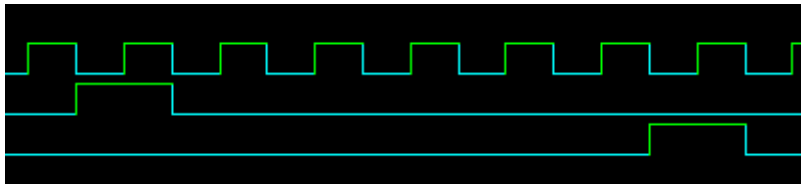


1<= Gap <= 5

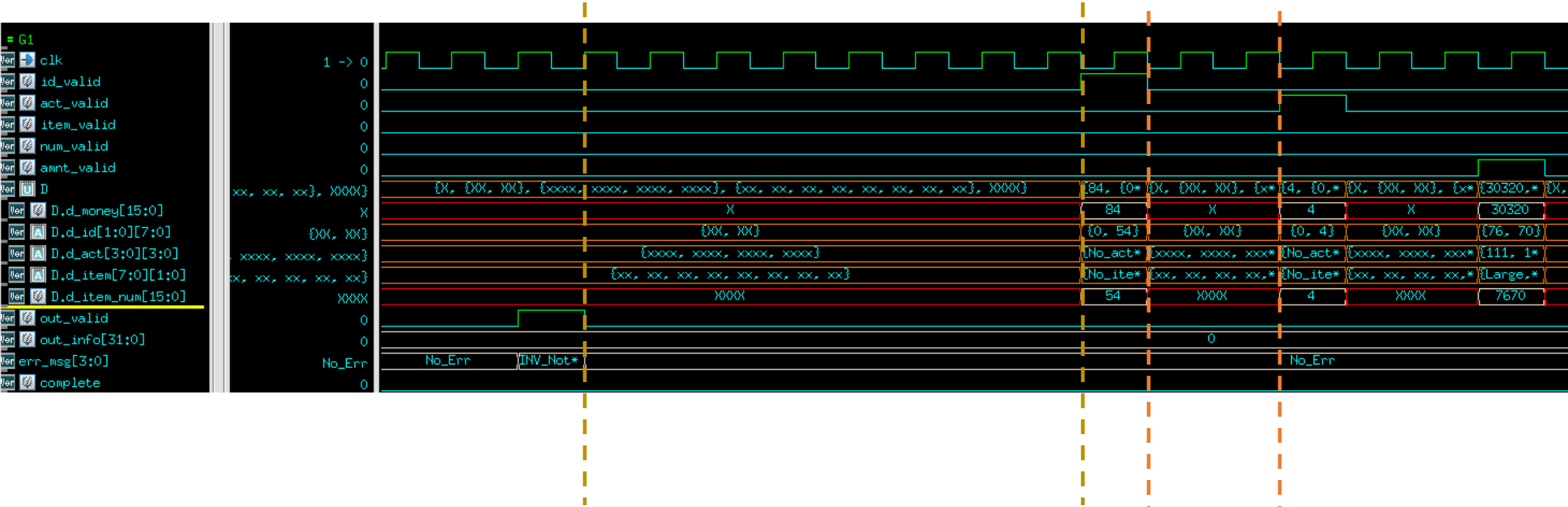
Order of input valid signal is fixed.

TA will not give the information of the delay between rst_n and the first input valid.

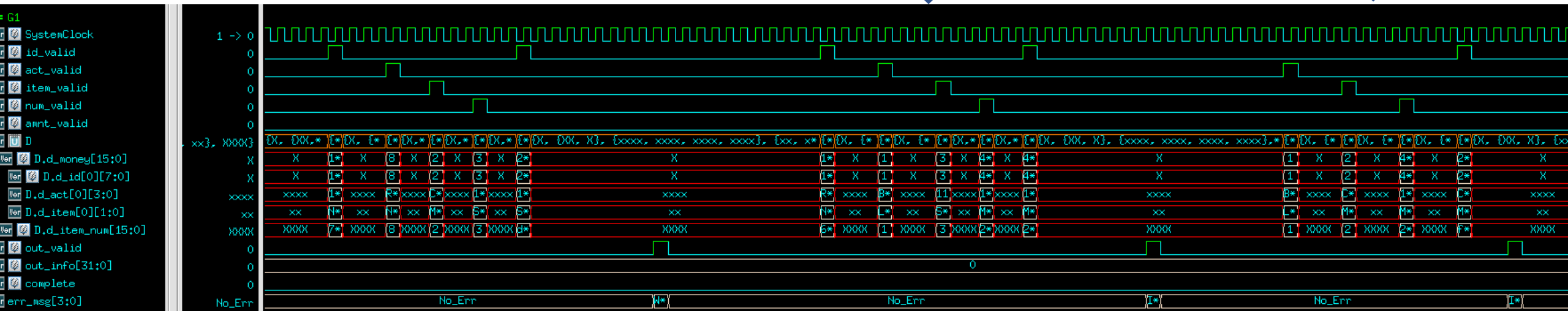
5 cycles



Next operation will be valid **2-10** cycles after out_valid fall.



If need…

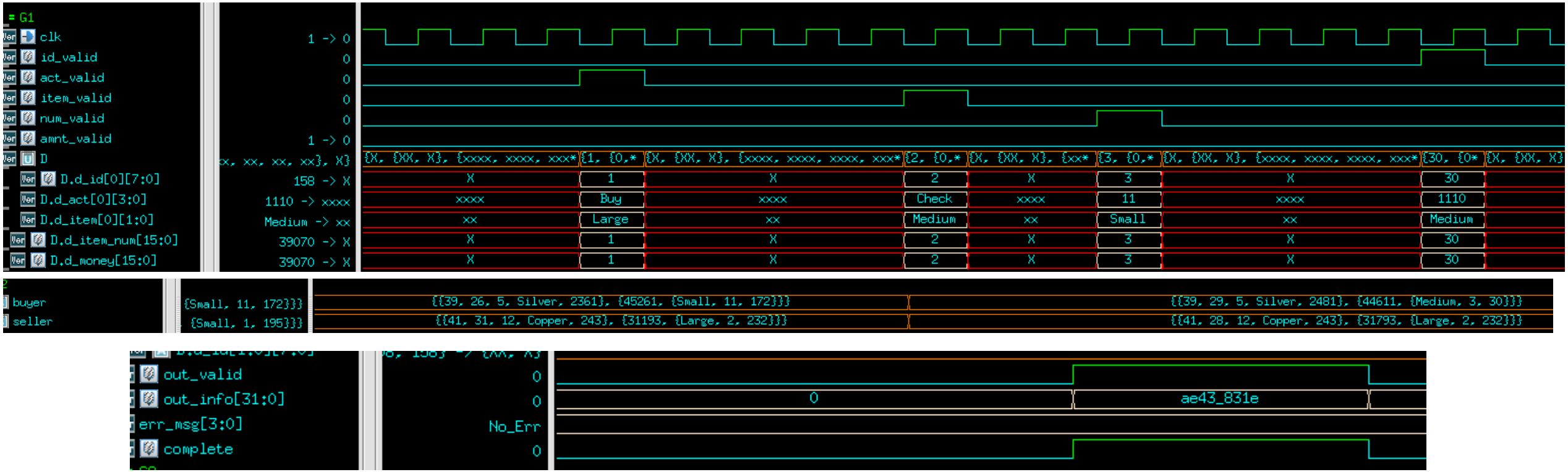


If PATTERN.sv want to change the current user, it will pull up the id_valid after out_valid pulling up. Otherwise, the user who need to do the given operations will be the same.

Example (Complete)

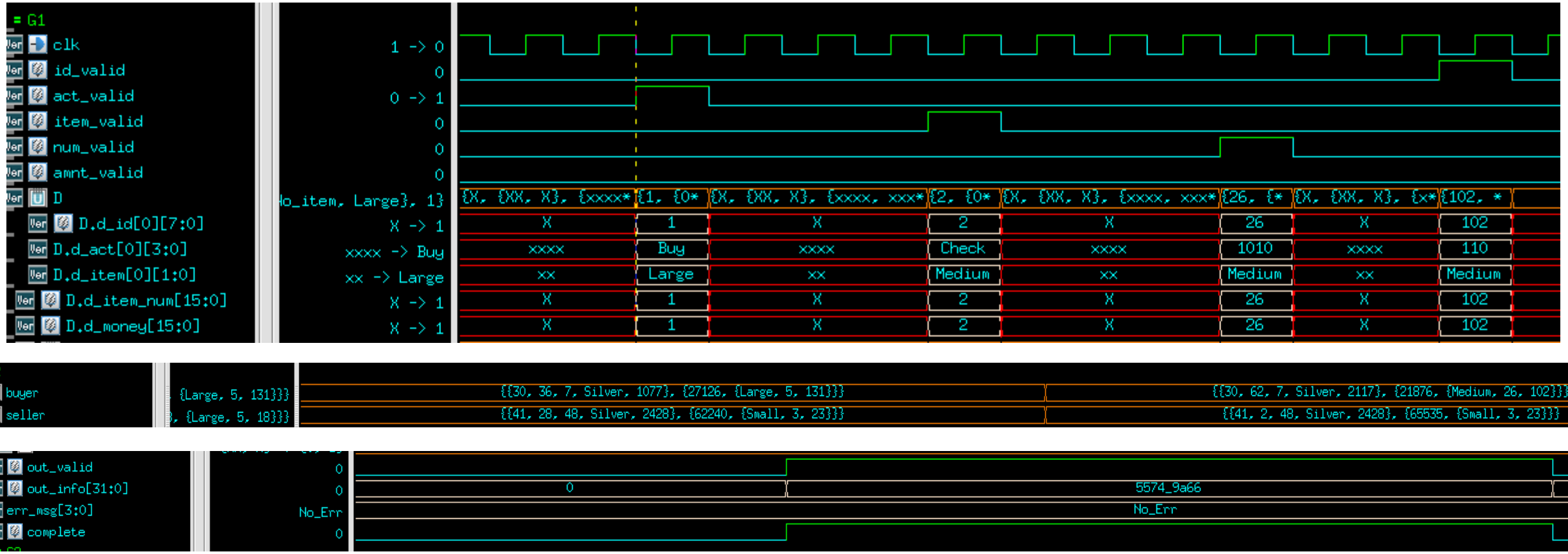
- Case 1 – Buy
- Case 2 – Buy (Seller’s wallet is full)
- Case 3 – Buy (User’s level upgrade)
- Case 4 – Check (User’s deposit)
- Case 5 – Check (Seller’s stocks)
- Case 6 – Deposit
- Case 7 – Return

Case 1 – Buy



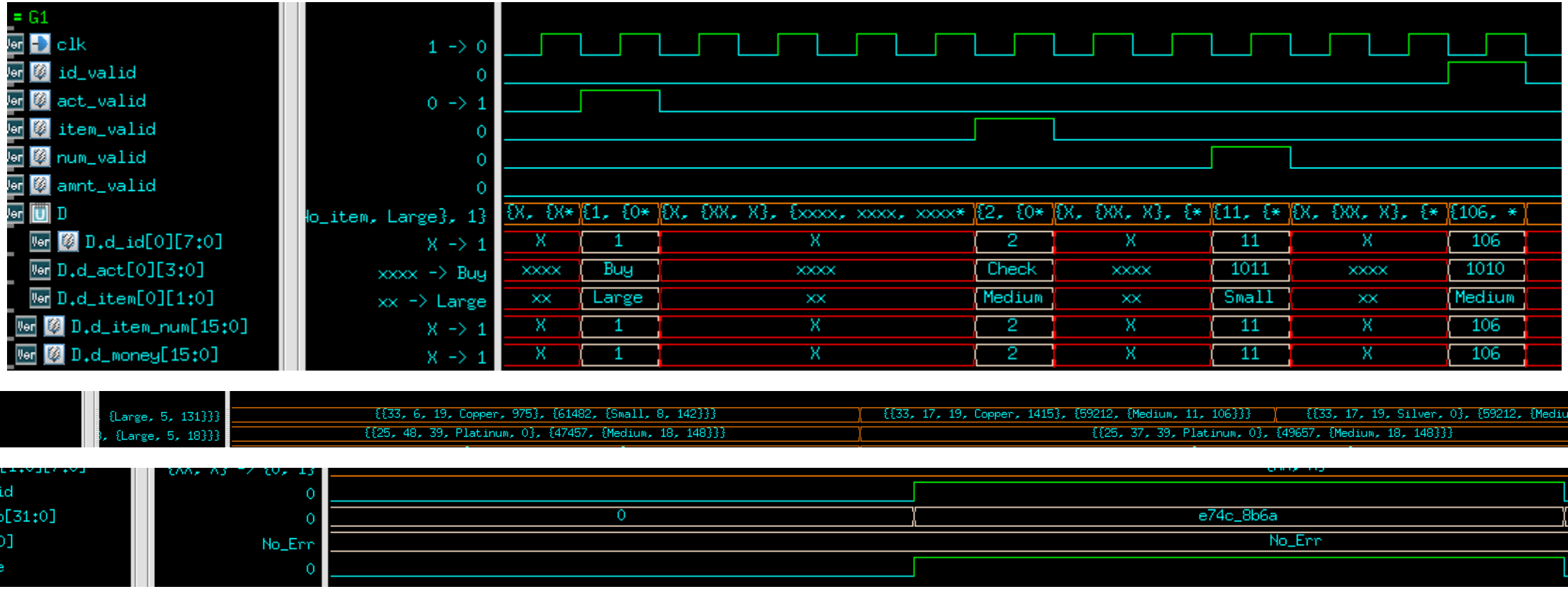
- Buyer means the current user. Seller is who sell the items to user.
- User: After ‘Buy’, # of corresponding items will increase and also the deposit will decrease. The shopping history and exp will be updated.
- Seller: After ‘Buy’, # of corresponding items will decrease and also the deposit will increase.

Case 2 – Buy (Seller’s wallet is full)



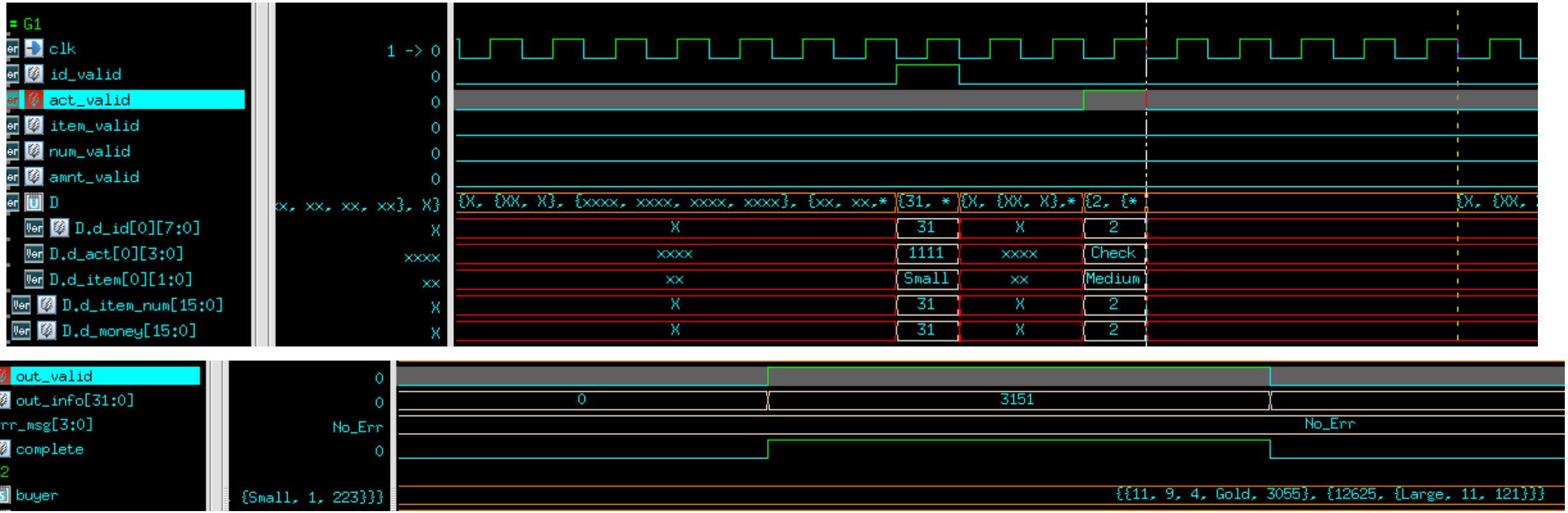
- When the money earned plus the money already owned exceeds 16’d65535 dollars, the seller’s deposit will be locked in 16’d65535, no matter how much money is earned.

Case 3 – Buy (User's level upgrade)



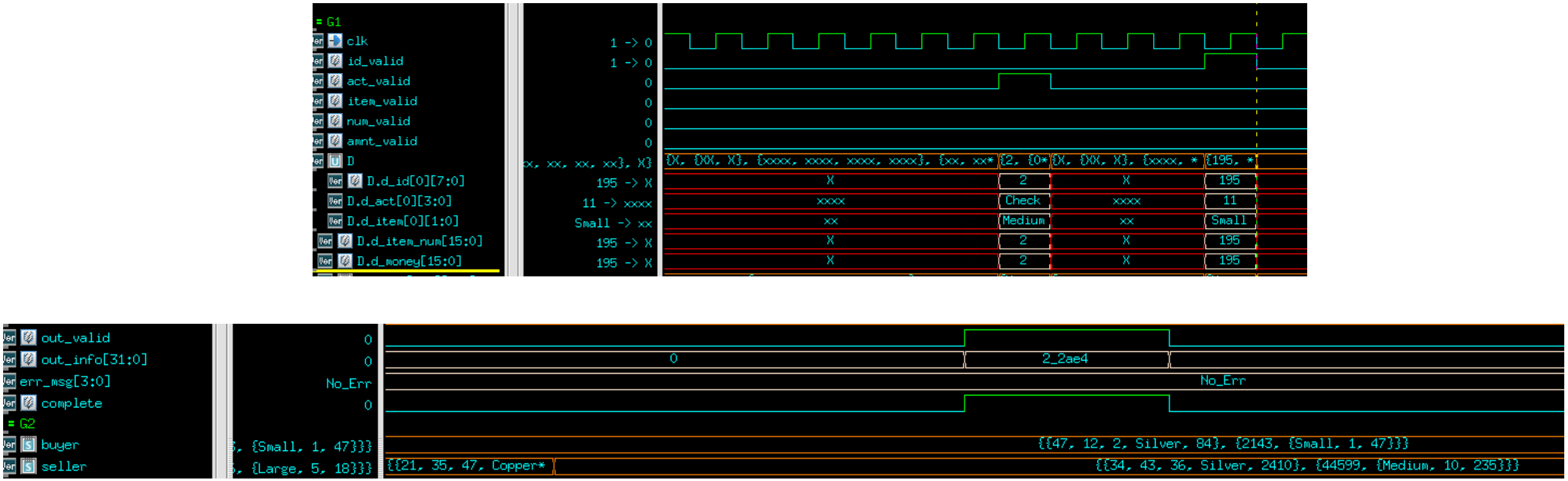
- When user's exp reaches the required value to upgrade, it need to update the user's level and exp will be cleared.

Case 4 – Check (User’s deposit)



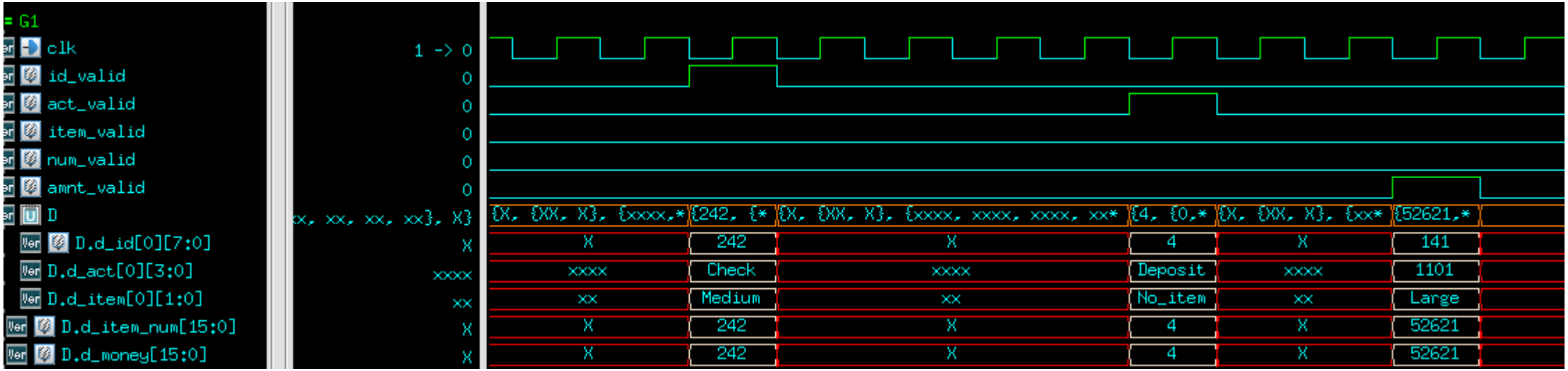
- If id_valid not be valid after 5 cycles, it means need to output user’s deposit.

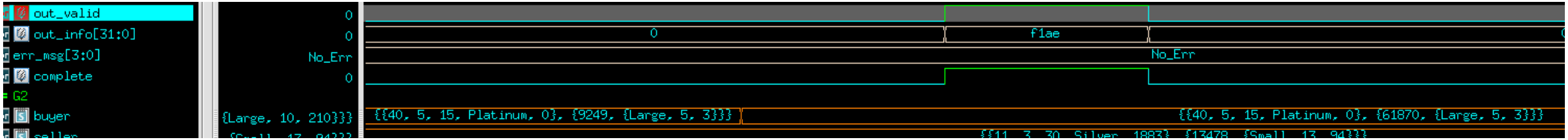
Case 5 – Check (Seller’s stocks)



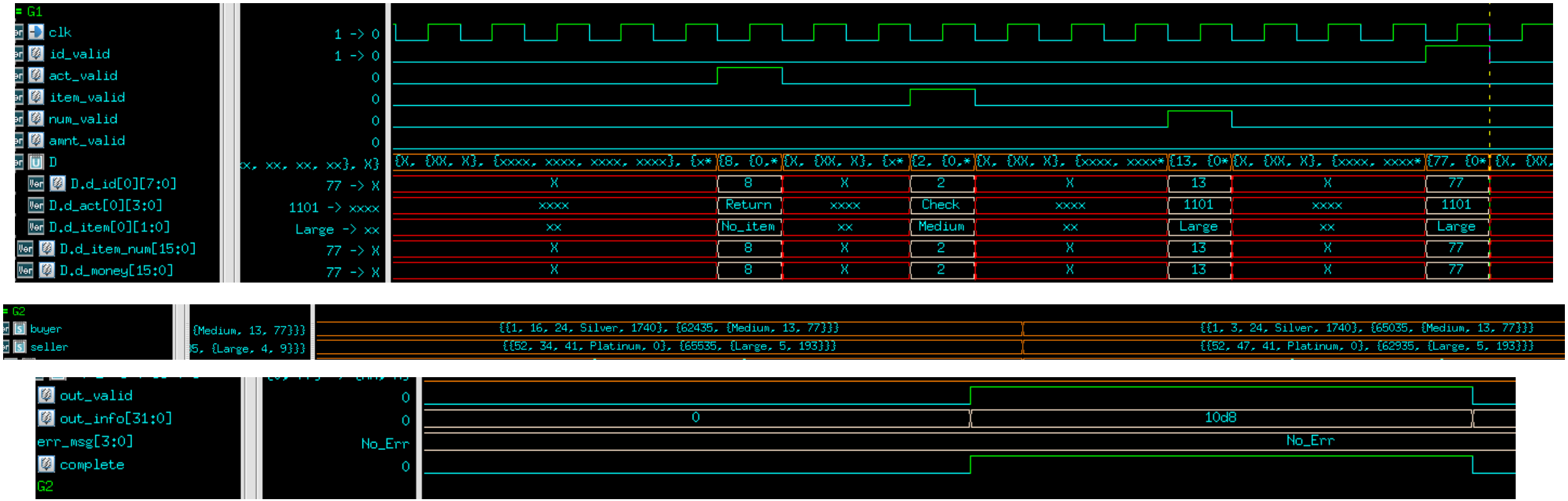
- If want to check the inventory of other stores, id_valid will be valid for 1-5 cycles after the act_valid fall.

Case 6 – Deposit





Case 7 – Return

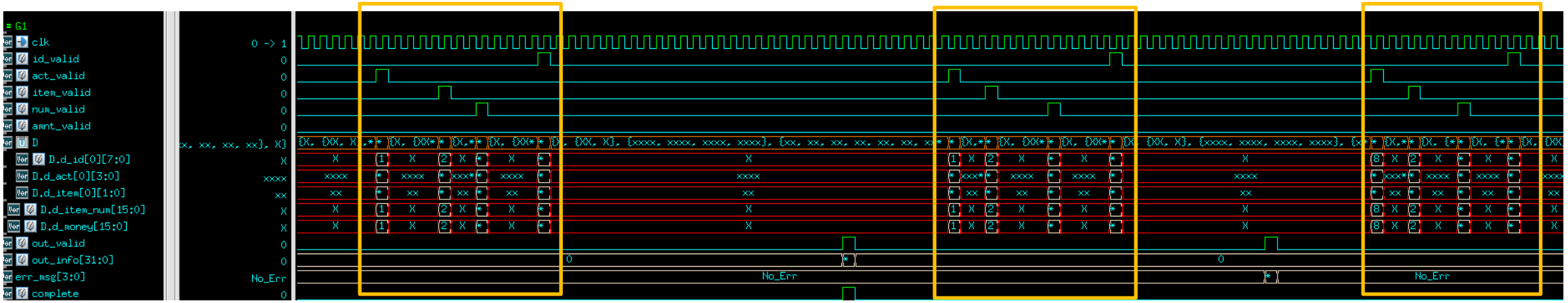


- User: After ‘Return’, # of corresponding items will decrease and also the deposit will increase.
- Seller: After ‘Return’, # of corresponding items will increase and also the deposit will decrease.

Case 7 – Return (Cont.)

Step2. Others operation

Step3.“Return”



Step1.“Buy”

Successful!

Fail!

- Step1. User A buys some items from User B, and it does successfully.
- Step2. Then, User A does another operation, however, it fails.
- Step3. User A wants to “Return” the items to User B and the given input matches with shopping history. It can be returned successfully.

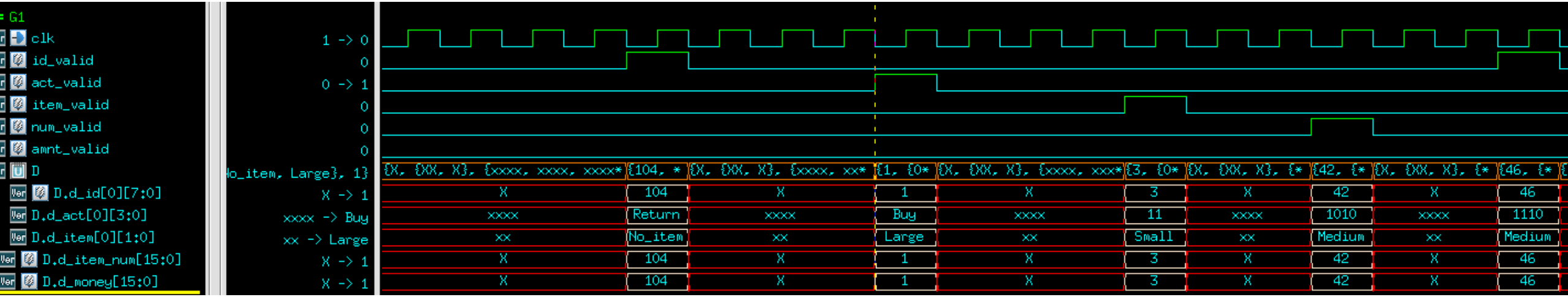
Hint: A situation can also be returned successfully.

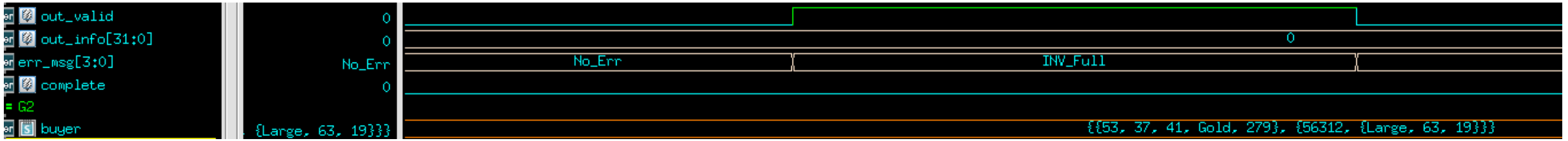
- Step1. User A buys some items from User B, and it does successfully.
- Step2. The PATTERN.sv change the current user into User C. User C does some operations which are not correlated with User A or B, such as buying some items from User A or B.
- Step3. The PATTERN.sv change the current user into User A. User A does another operation, however, it fails. Then, User A wants to “Return” the items to User B, it can be returned successfully.

Example (Error)

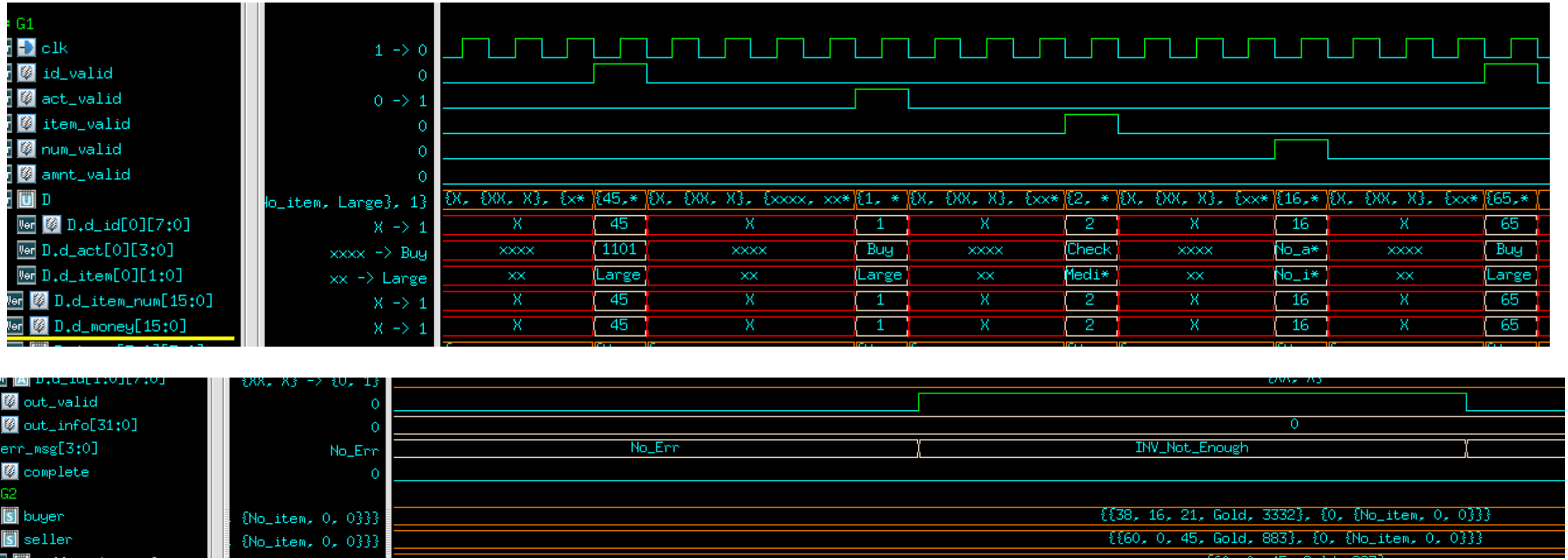
- Case 8 – Buy, but user's inventory is full
- Case 9 – Buy, but seller's inventory is not enough
- Case 10 – Buy, but out of money
- Case 11 – Deposit, but wallet is full
- Case 12 – Return, but wrong operation(not be made immediately after purchase)
- Case 13 – Return, but wrong operation(not the most recent buyer)
- Case 14 – Return, but wrong seller ID
- Case 15 – Return, but wrong item ID
- Case 16 – Return, but wrong number

Case 8 – Buy, but user’s inventory is full

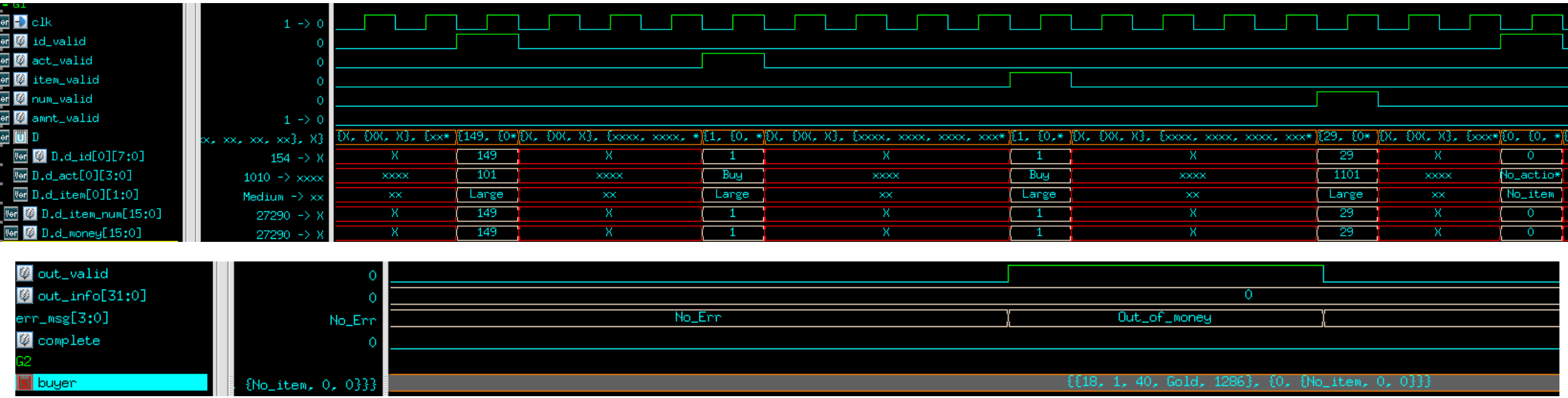




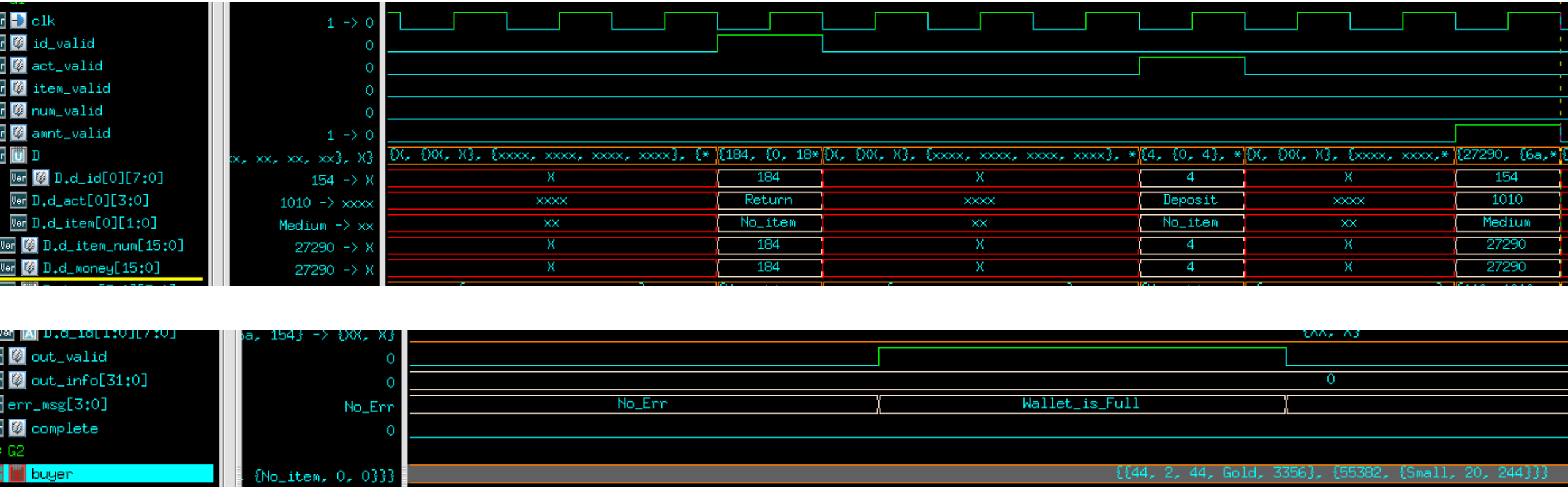
Case 9 – Buy, but seller’s inventory is not enough



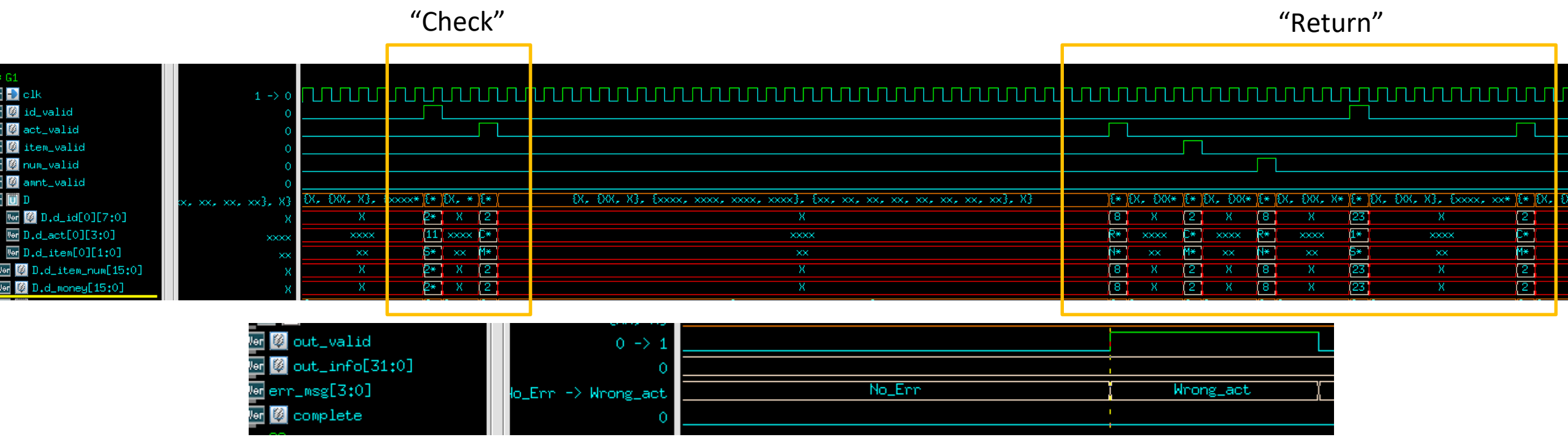
Case 10 – Buy, but out of money



Case 11 – Deposit, but wallet is full



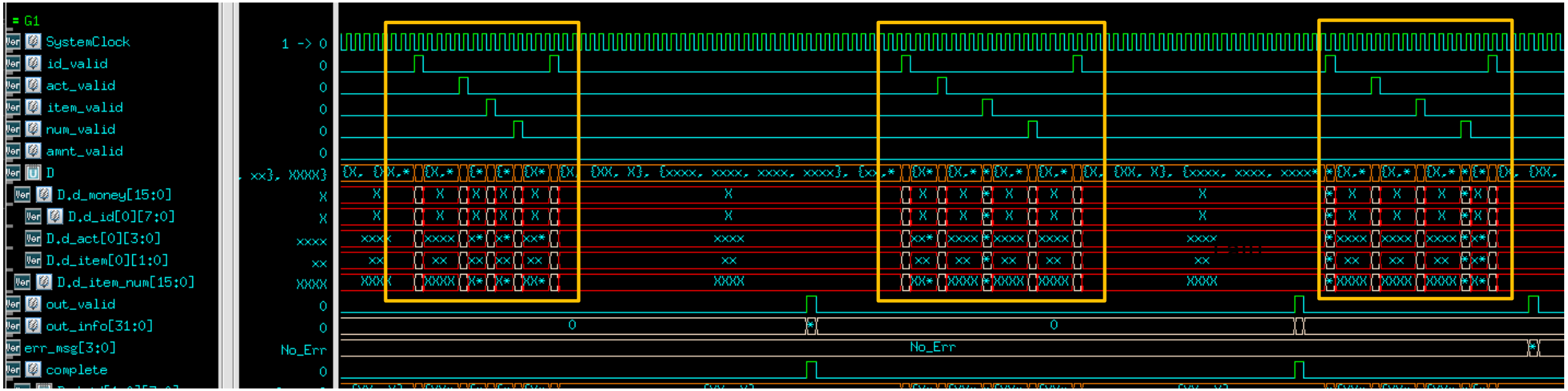
Case 12 – Return, but wrong operation(not be made immediately after purchase)



- The user has done “Check” operation. Therefore, will need to output “Wrong act” after get “Return” operation.
- If A buy items from B and B do some operations, it will also need to output “Wrong act” when A want to “Return”.

Case 13 – Return, but wrong operation(not the most recent buyer)

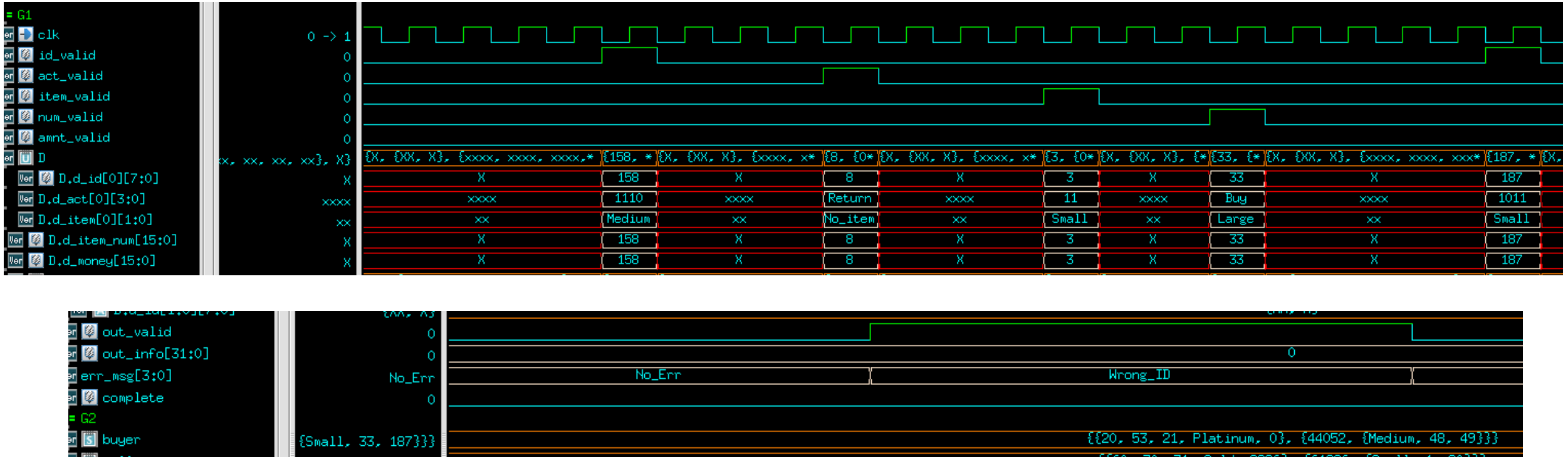
Step1 Step2 Step3



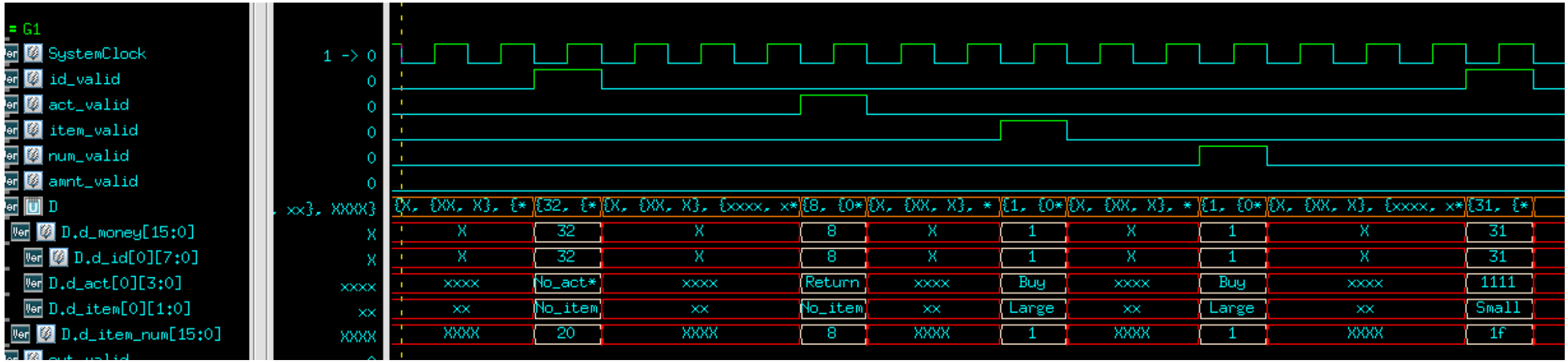
Successful! Successful! Fail!

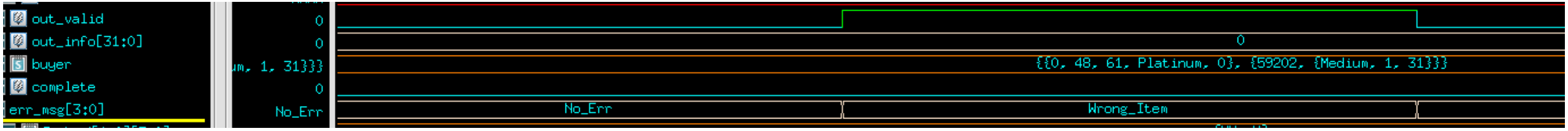
- Step1. User A buys some items from User B, and it does successfully.
- Step2. Then, User C buys some items from User B, and it does successfully.
- Step3. User A wants to “Return” the items to User B, it will be taken as “Wrong act”.

Case 14 – Return, but wrong seller ID

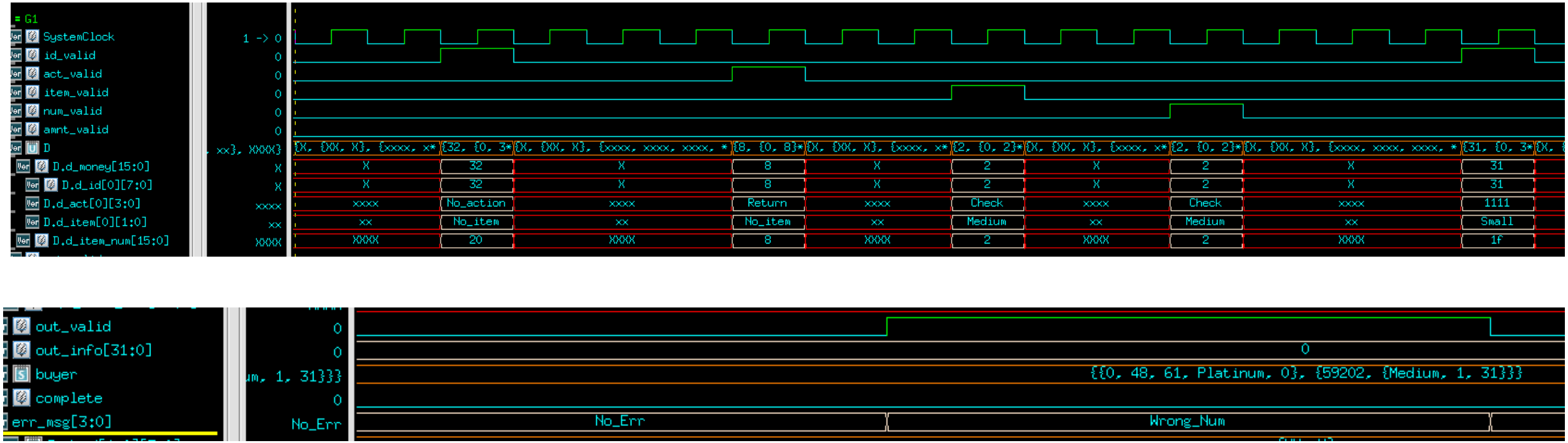


Case 15 – Return, but wrong item ID





Case 16 – Return, but wrong number



GOOD LUCK!