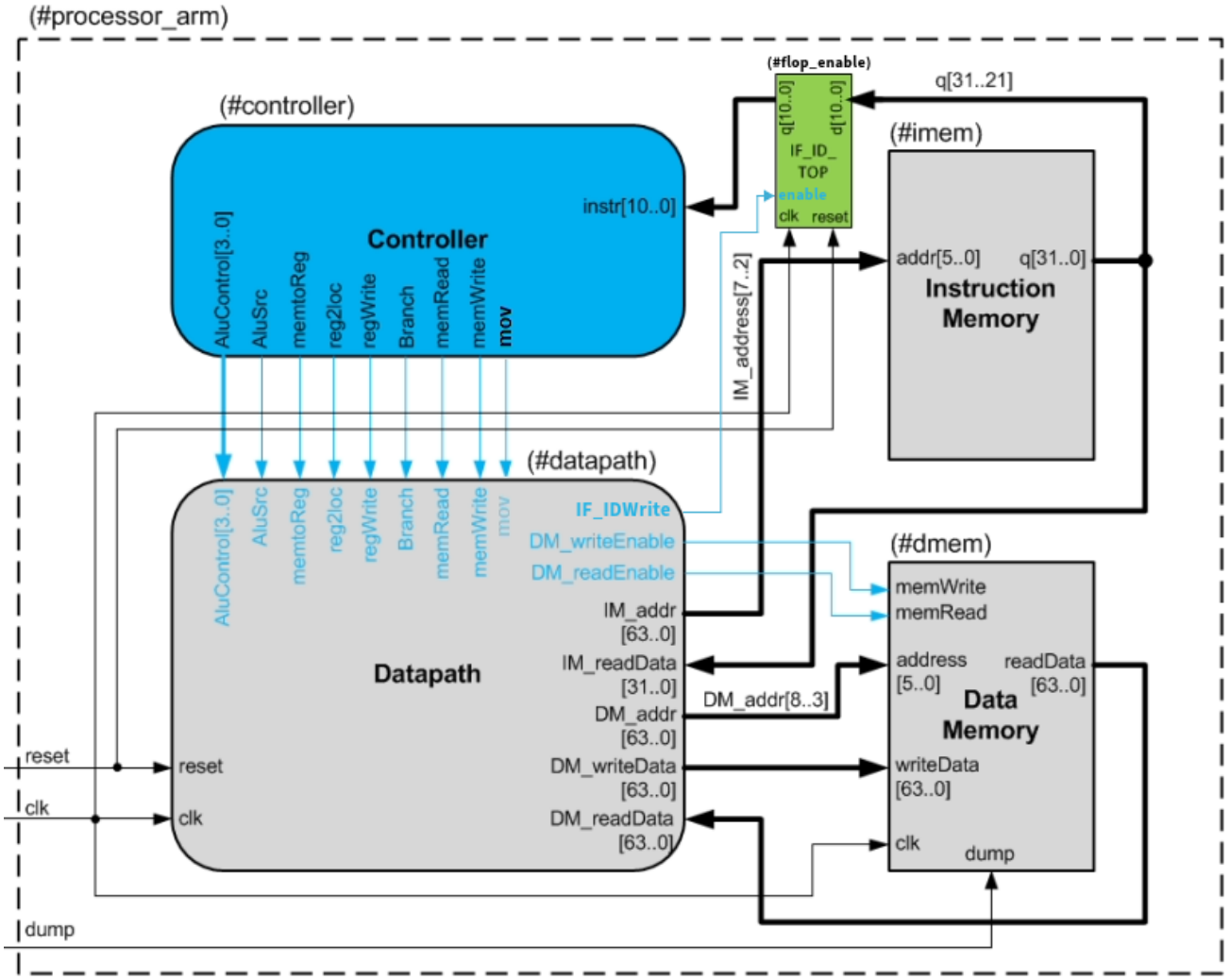
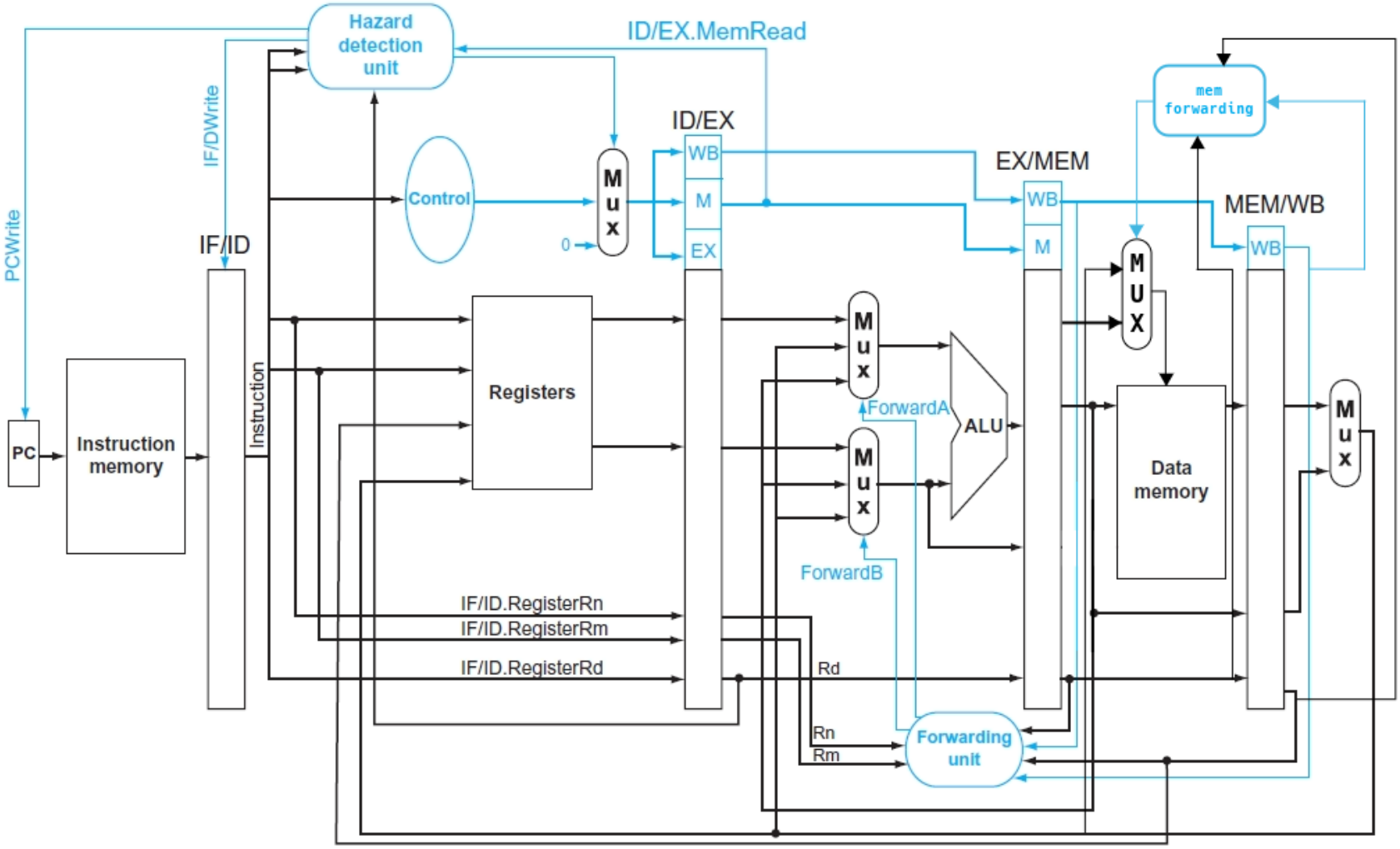
INFORME LAB 1
ARQUITECTURA DEL COMPUTADOR

Integrantes

- Cristian Ariel Muñoz
- Santiago León Torres Banner
- Damián Feigelmüller

Diagramas:




Hazard detection unit
ID/EX.MemRead
IF/DWrite
PCWrite
ID/EX
EX/MEM
MEM/WB
mem forwarding
Control
M u x
0
WB
M
EX
IF/ID
Instruction
PC
Instruction memory
Registers
ALU
Data memory
ForwardA
ForwardB
IF/ID.RegisterRn
IF/ID.RegisterRm
IF/ID.RegisterRd
Rd
Rn
Rm
Forwarding unit

CASO STALL

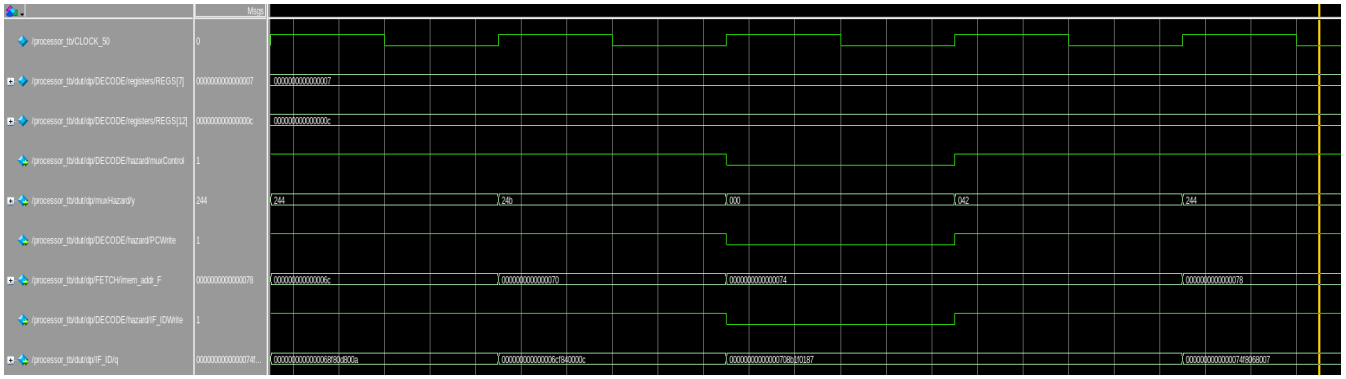


FIG. 1 Caso de *stall*, los valores están en hexadecimal

En la imagen observamos el comportamiento de esta parte del código, donde hay dependencia de datos entre LDUR y ADD en el registro X12:

LDUR X12, [X0, #0]
ADD X7, X12, XZR

Para que el LDUR pueda guardar en el registro el valor que está en memoria tiene que pasar por la etapa Memory. Provocando que la instrucción ADD tenga que realizar un *stall* para poder utilizar el valor del registro con *forwarding*.

Podemos notar en la captura (FIG. 1) que cuando sucede un *stall* (el registro a modificar por el LDUR es el mismo que va a utilizar el ADD), se ponen las señales de *PCWrite*, *IF_IDWrite* y *muxControl* en 0, esto provoca que el PC y el *IF_ID* no cambien de valor, y que las señales de control tomen valor “0” a partir del ciclo EX. Se visualiza en la captura (FIG. 1) que las señales que salen cuando el *muxControl* es “0”, son todas “0”, luego van a pasar por todas las etapas, comenzando en Execute.

CASO FORWARDING

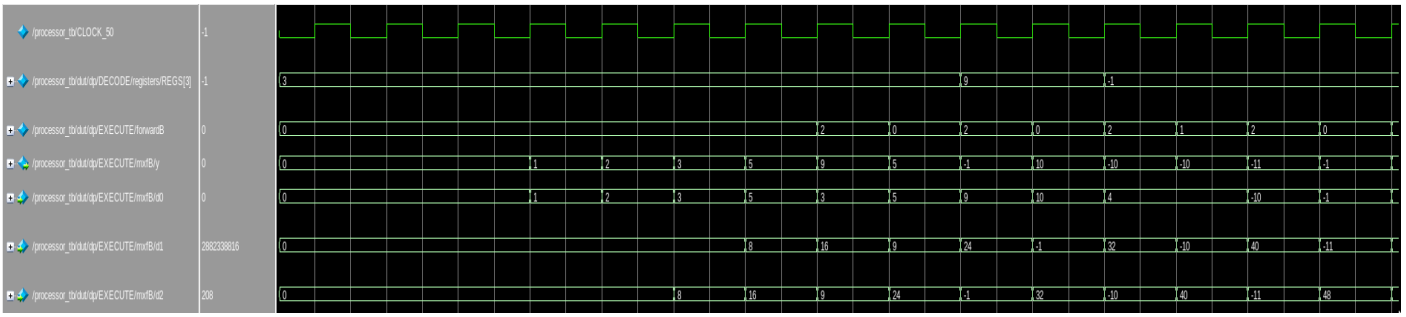


FIG. 2 Caso de *forwarding*, los valores están en decimal

En la imagen observamos el comportamiento de esta parte del código, donde hay dependencia de datos entre ADD y STUR en el registro X3:

ADD X3, X4, X5
STUR X3, [X0, #24]

Podemos notar en la captura (FIG. 2) que el valor de *ForwardB* cambia a 0b10, es decir que se modifica el valor del registro que está por guardarse en memoria. Que sea 0b10 significa que el valor que necesitamos es el que sale de la pipe **EX/MEM**. Es el valor que sale por el **mux4** (mxB/*y*).
Se puede observar que hace un *forwarding* porque utiliza el valor actualizado de X3 antes de que sea escrito en memoria.