Cache Fetch

TABLE OF CONTENTS

1. QUICK REVIEW ON MEMORY

The memory we have implemented so far:

- Takes a 32-bit address and a 1-bit clock as inputs
- Produces 128 bits as output
- It takes 5 clocks to produce the output

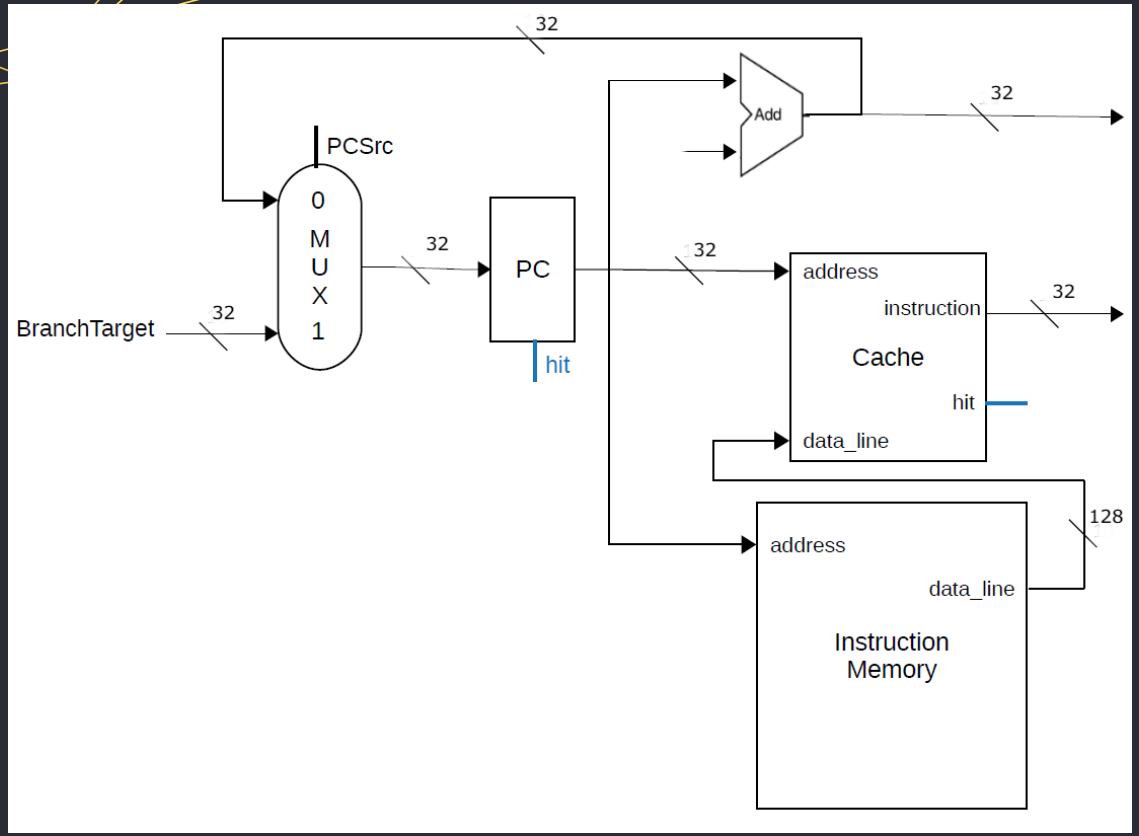
2. WHY WE USE CACHE

- As previously stated, memory requires some time to produce data and put it on output lines. (We simulated this by outputting data on the 5th clock)

- The *Principle of Locality* suggests that items whose addresses are near one another tend to be referenced close together in time (Spatial) and recently accessed items are likely to be accessed in the near future (Temporal).

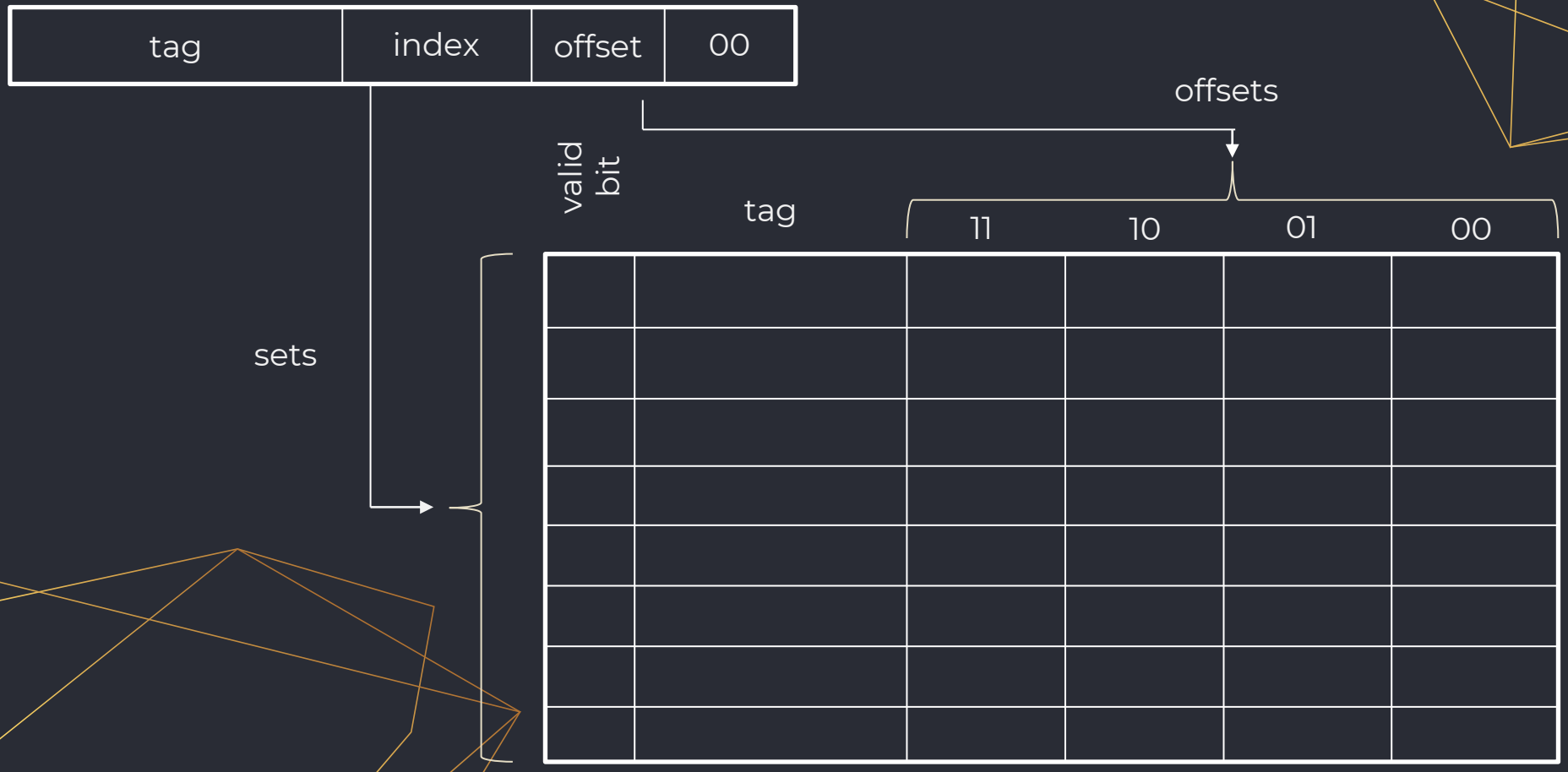
- Cache is used to hold data for addresses which are likely to be accessed in a near future.

3. HOW TO IMPLEMENT CACHE



Note that the hit bit is used to determine the right time for latching the PC.

3. HOW TO IMPLEMENT CACHE (Cont'd)



4. REVISION ON PC REGISTER

So, an instruction is output from the cache on the positive edge of clock. Next instruction address is also being latched on the positive edge of clock...

YOU SEE WHERE I'M GOING WITH THIS?

Therefore, latch the new address on the negative edge of the clock.

5. IMPLEMENT FETCH MODULE

Finally, it's time to instantiate the modules and connect them together. Modules which need to be instantiated and connected are as follows:

Multiplexer:

- Inputs: **PC**(32 bits), **Branch Target**(32 bits, set to 0 for now), **PC Source**(1 bit)
- Outputs: **Address**(32 bits, chosen among PC and Branch Target)

PC Register:

- Inputs: **Address**(32 bits, the output of the multiplexer), **Hit** (1 bit, indicating whether the cache has been hit or not), **Clock**(1 bit)
- Outputs: **Address**(32 bits)

Memory:

- Inputs: **Address**(32 bits), **Clock**(1 bit)
- Outputs: **Data Line**(32 bits)

Cache:

- Inputs: **Address**(32 bits), **Data Line**(128 bits), **Clock**(1 bit)
- Outputs: **Instruction**(32 bits), **Hit**(1 bit)

5. IMPLEMENT FETCH MODULE (Cont'd)

Fetch:

- Inputs: **Clock**(1 bit), **Branch Target**(32 bits, set to 0 for now), **PC Source**(1 bit)
- Outputs: **Next PC**(32 bits), **Instruction**(32 bits), **Hit**(1 bit)