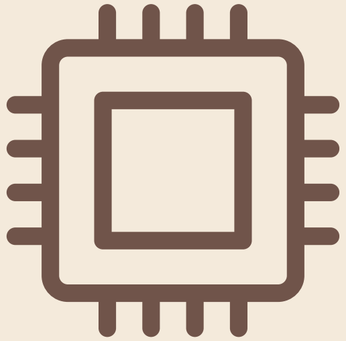
PROCESSOR DESIGN WORK

SPRING 2024

Fatima Farooq

Stages in PROCESSOR DESIGN

RTL Design



Verification

Syntehsis

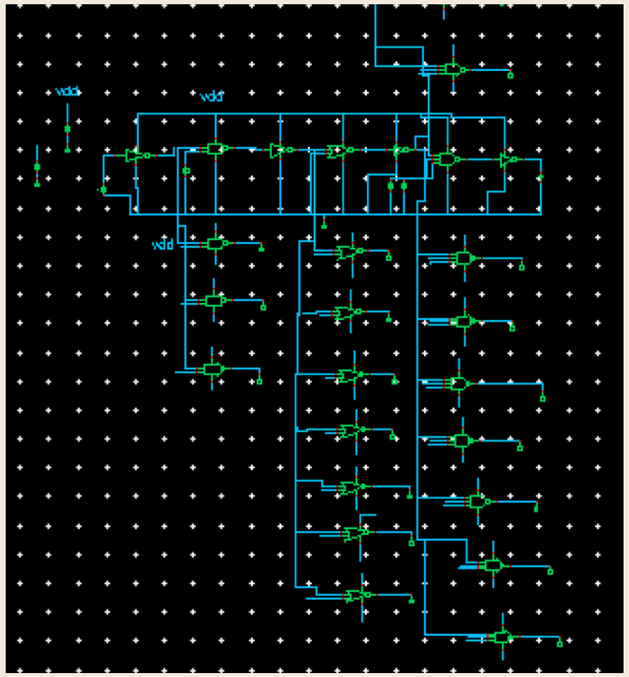
Layout

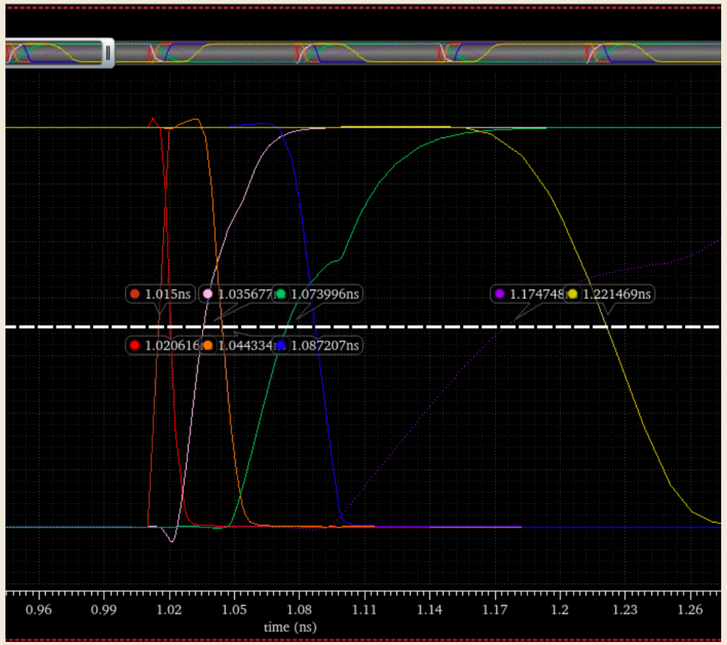
tapeout & Validation in real silicon

Tools

- Tool chain: collection of tools that sequentially work
- Cmake: build automation tool
    - describes the build process using a CML
- Virtuoso: suite of hardware deisgn tools

```
cmake_minimum_required(VERSION 3.24)

if(NOT DEFINED CMAKE_TOOLCHAIN_FILE)
  include(FetchContent)
  FetchContent_Declare(
    vcpkg
    GIT_REPOSITORY https://github.com/microsoft/vcpkg.git
    GIT_TAG master
    GIT_SHALLOW TRUE
  )
  FetchContent_MakeAvailable(vcpkg)
  set(CMAKE_TOOLCHAIN_FILE
    ${vcpkg_SOURCE_DIR}/scripts/buildsystems/vcpkg.cmake
    CACHE FILEPATH "Vcpkg toolchain file"
  )
  set(VCPKG_ROOT_DIR ${vcpkg_SOURCE_DIR} CACHE PATH "Vcpkg Root Directory")
endif()

add_custom_target(UpdateVcpkgBaseline
  ${VCPKG_ROOT_DIR}/vcpkg x-update-baseline
)

project(week-four-lab VERSION 1.0.0)
find_package(nyu-cmake CONFIG REQUIRED)
add_library(lab4 INTERFACE)
add_subdirectory(rtl)

if(NYU_BUILD_TESTS)
  enable_testing()
  add_subdirectory(dv)
endif()
```





Design Verification

- RTL simulation
- Gate-level simulation
- Testing RTL modules
- Clear requirements of the behavior of the system
- Basic functionality & edge cases

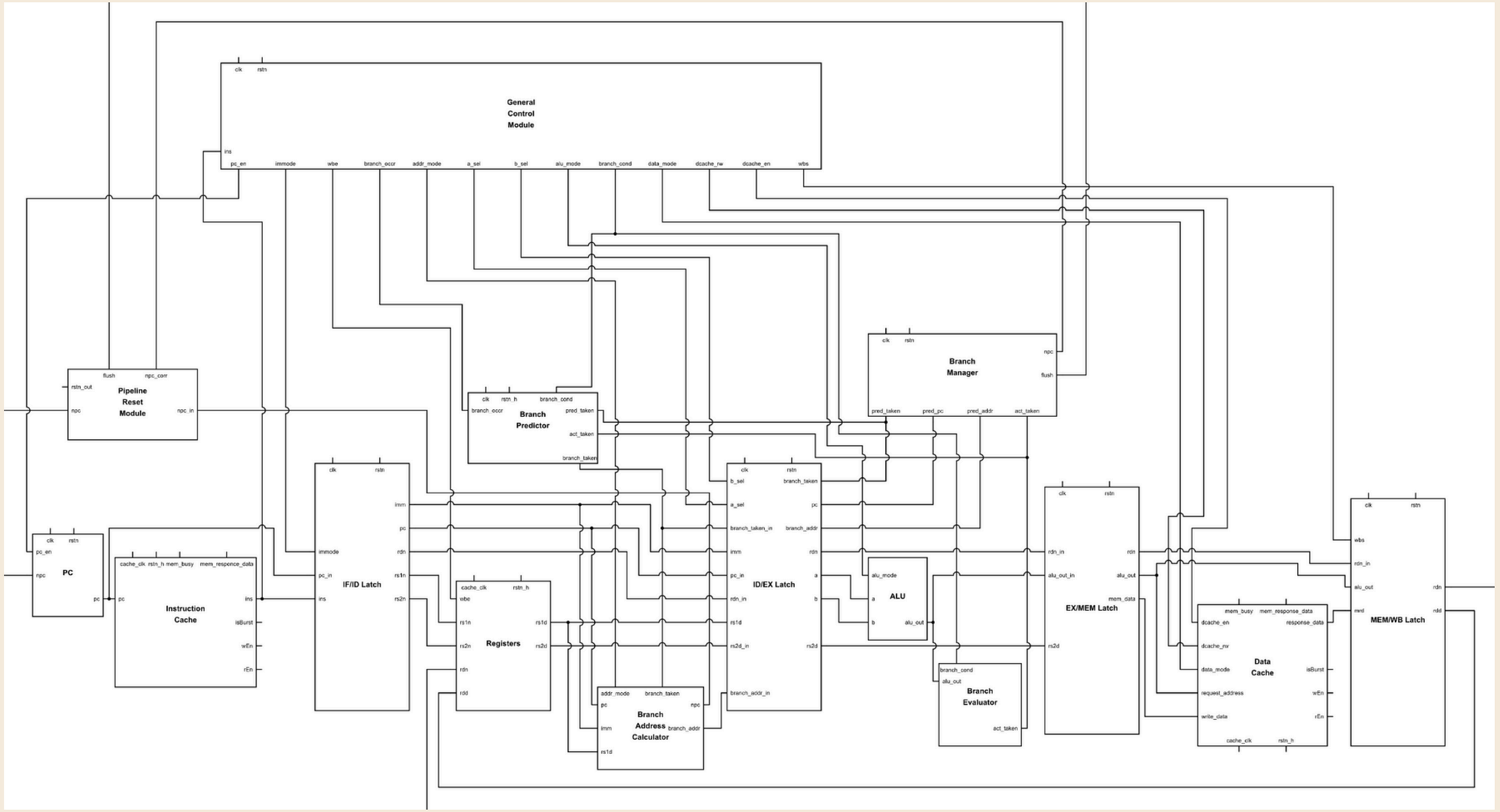
```
    model.clk = 0; // getting to negative edge
    model.reset = 1;
    model.init = num;
    model.eval(); // out now has ~ num

    uint16_t last = ~num;

    for (int i=0; i <100; i++)
    {
    // getting negative edge of clock
    model.clk = 1;
    model.reset = 0;
    model.eval();
    model.clk = 0;
    model.reset = 0;
    model.eval();

    // check
    if (model.out != fib(last))
    {
      return false;
    }

    last = model.out; // update last
    }

return true;

}

int main() {

  REQUIRE(test(5)==true);
  REQUIRE(test(999)==true);
  REQUIRE(test(503)==true);
  REQUIRE(test(1001)==true);
  REQUIRE(test(5167)==true);
  REQUIRE(test(21199)==true);
  REQUIRE(test(65535)== true);

return 0;
```

CPU core

- Control Unit (CU)
- Fetch and Decode Units (FDU)
- AGUs (Adress Generation Unit)
- L1 cache
- Translation Lookaside Buffer (TLB)
- Load-Store Unit (LSU)
- FPU(s) (Floating-Point Unit)
- SIMD Unit(s)
- Instruction Pipeline
- Branch Prediction Unit
- Multiple registers

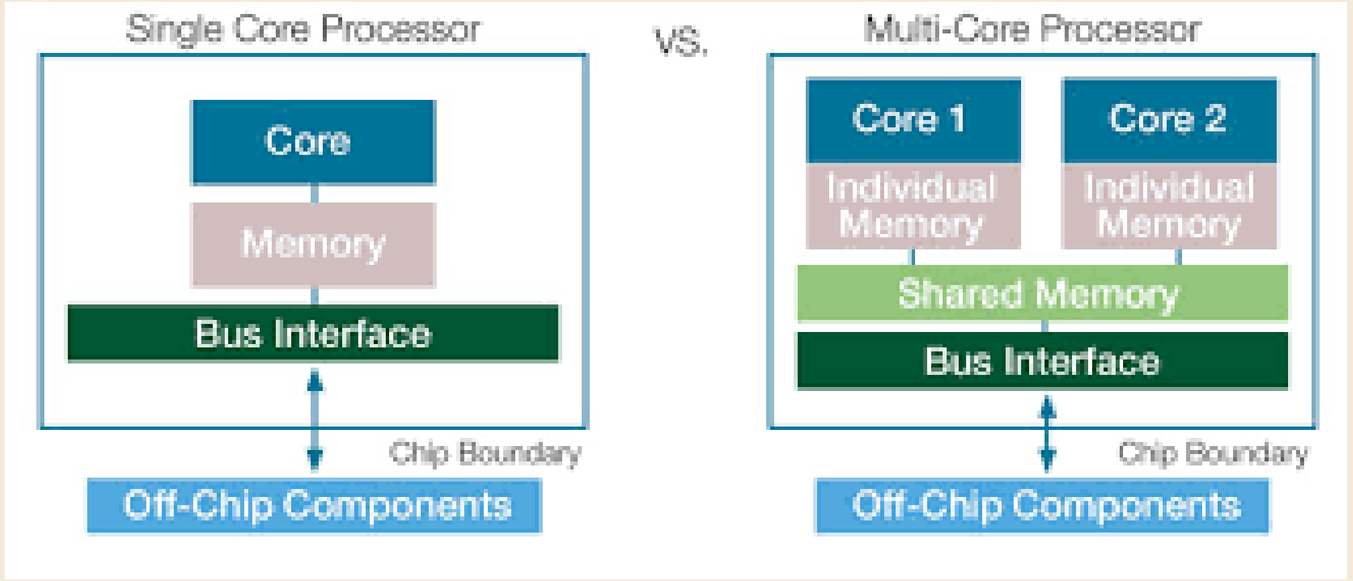
NYU core

- Latch modules
- Pipeline module



Multiple cores

- Parallelization of different processes
- Multi-threaded processes
- Communication between different caches
- Increased cache coherency performance
- Increased Power consumption
- Thread scheduling can be challenging



References

- https://medium.com/nerd-for-tech/multi-core-processors-53ee2899f90f
- https://en.wikipedia.org/wiki/Multi-core_processor#Advantages
- https://medium.com/@razvanbadescu/anatomy-of-a-cpu-bc02cd950cca
- https://www.redhat.com/sysadmin/cpu-components-functionality#:~:text=Core%20%2D%20A%20core%20is%20the,into%20a%20single%20physical%20package.