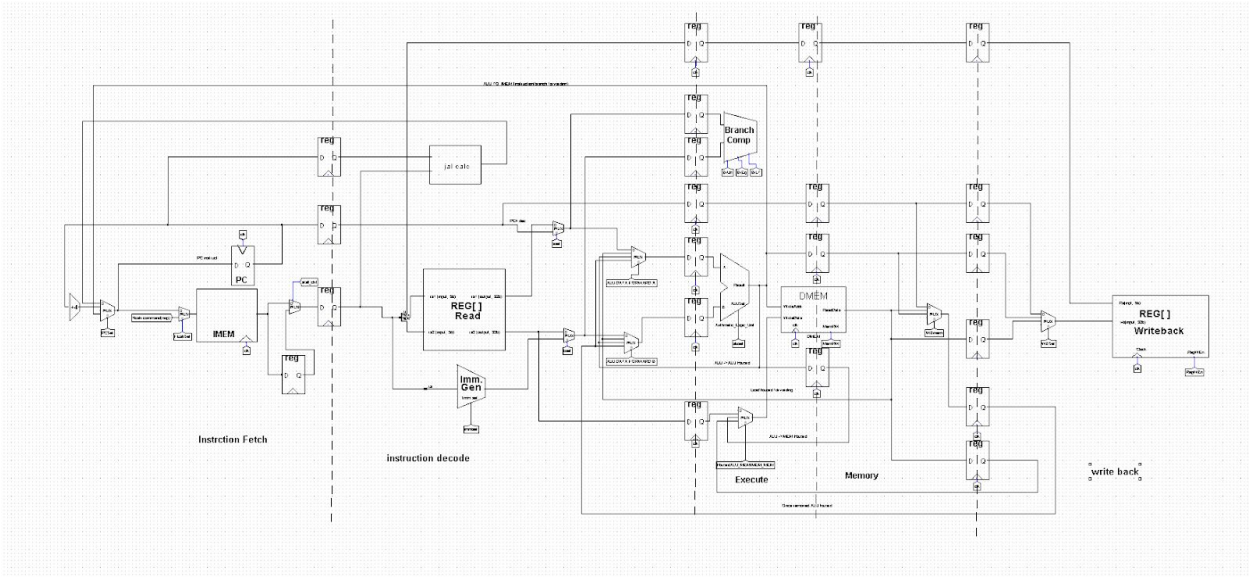
Final Report

EECS 151 ASIC Lab

Ryan Thornton and Kevin Jung

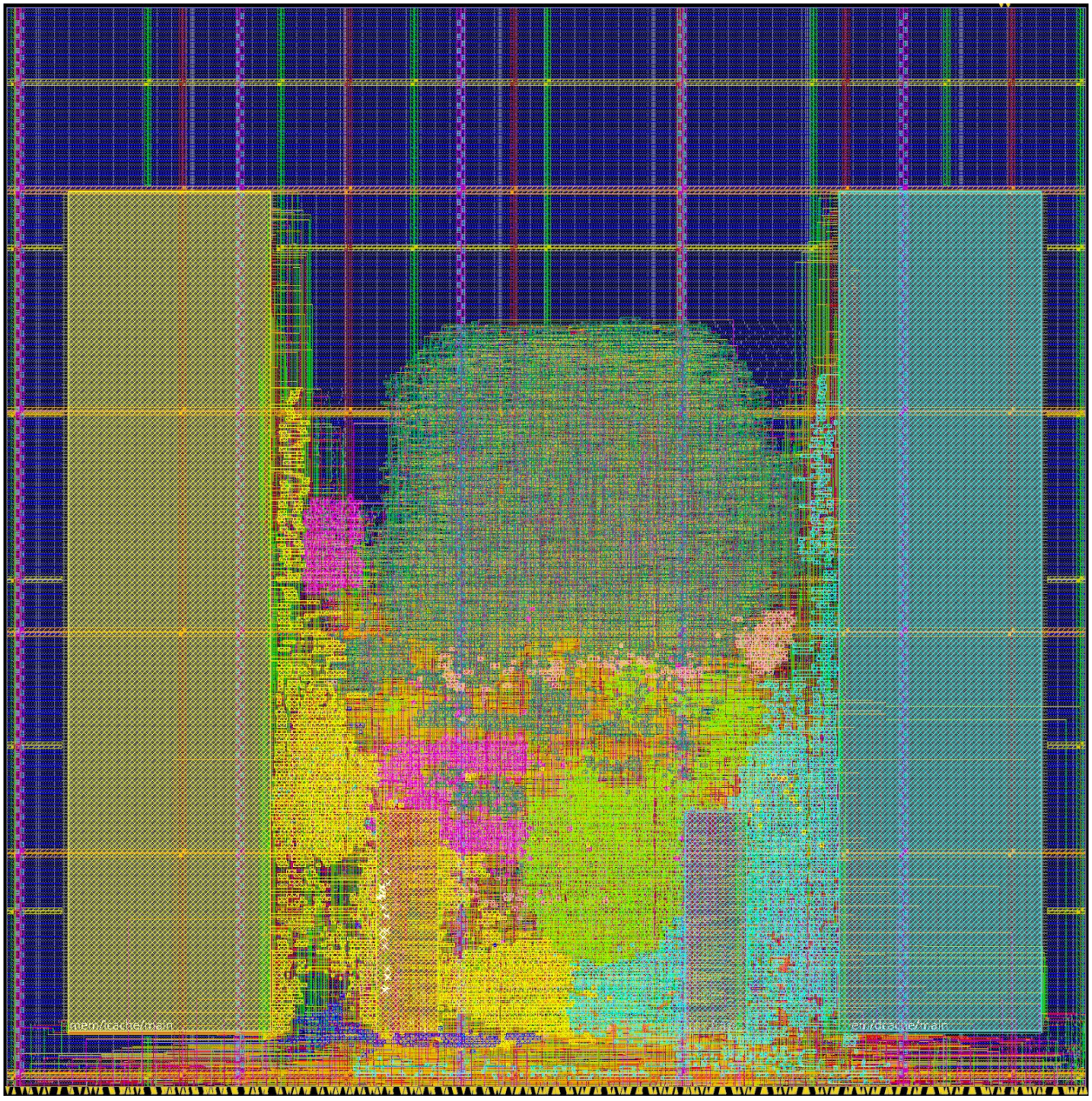
1) Final Pipeline Diagram:



2) Post-Synthesis Critical Path:

The post-synthesis critical path (with a target clock period of 0.6ns) is from the instruction IF/ID pipeline register through the JAL-specific adder through the PC selector mux to the icache request address buffer register. This route has a positive slack of 16ps.

3) Final Floorplan:



Yellow is instruction cache
Light blue is data cache
Red (barely visible, inside data cache) is memory arbiter
Dark blue at bottom is writeback stage
Pink is instruction fetch stage
Darker green is instruction decode stage
Light green is execute stage
Orange is memory stage
External facing modules (i$ and d$) are as close to the pins as possible to try to minimize delay

4) Post-PAR Critical Path:

The reported post-PAR critical path is from cpu/Bsel_pip/q_reg[4]/CLK to cpu/id_stage/haz/rd_log_ex/CLKGATE_RC_CG_HIER_INST6/RC_CGIC_INST/ENA.

This path travels through the ALU, into the data cache, through the data cache's stall logic, into the hazard controller in instruction decode. Interestingly, in this iteration, this post-route path fails the synthesis target clock of 600ps, with a 0.721ps negative slack. However, optimizing this path to meet the timing requirements has absolutely no impact on the clock speed that the CPU can be simulated at.
In reality, Innovus shows, with the command **`report_timing -unconstrained -from mem/dcache/mem_req_valid -to mem_req_valid`**:

```
#############################################################
#  Generated by:      Cadence Innovus 18.10-p002_1
#  OS:                Linux x86_64(Host ID eda-6.EECS.Berkeley.EDU)
#  Generated on:      Mon May 11 22:01:13 2020
#  Design:            riscv_top
#  Command:           report_timing -unconstrained -from
mem/dcache/mem_req_valid -to mem_req_valid
#############################################################
Path 1:
                View: PVT_0P63V_100C.setup_view
     Startpoint: (R) mem/dcache/req_address/q_reg[21]/CLK
          Clock: (R) clk
          Endpoint: (R) mem_req_valid
          Clock:

                        Capture    Launch
     Src Latency:+      0.000      -85.256
     Net Latency:+      0.000 (I)   89.700 (P)
          Arrival:=     0.000      4.444

     Launch Clock:=     4.444
          Data Path:+   903.100

#-------------------------------------------------------------------
-----------------------------------------------------------------
-
# Timing Point                                    Flags  Arc
Edge  Cell                      Fanout     Trans      Delay   Arrival
#
     (ps)       (ps)       (ps)
```

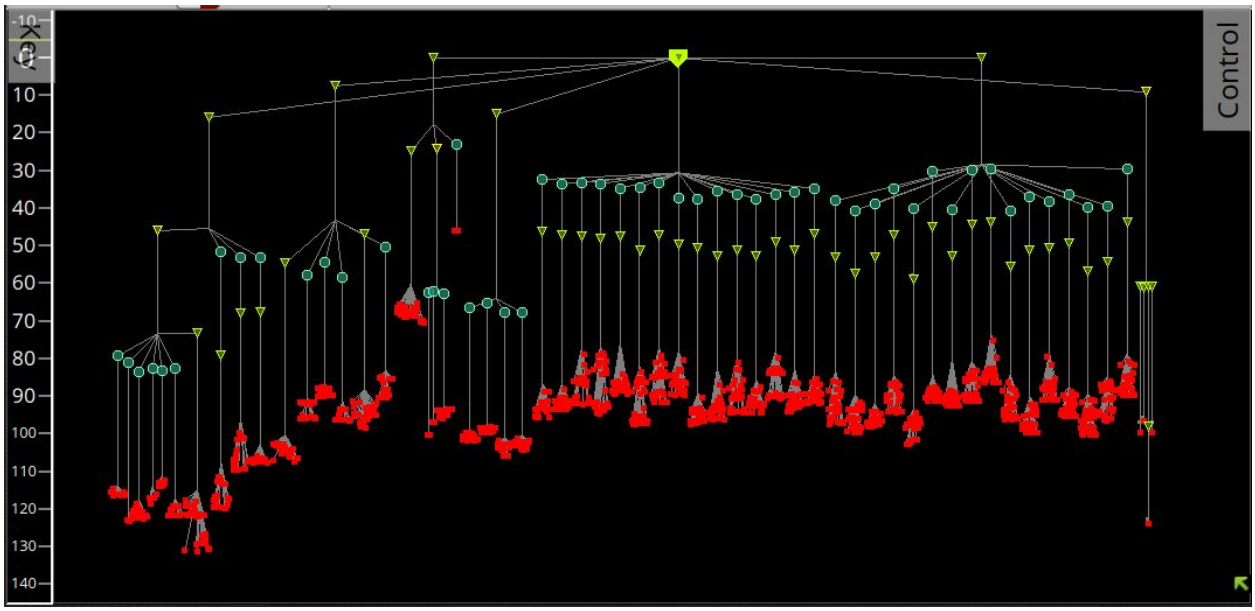
```
#-------------------------------------------------------------------
-------------------------------------------------------------------
-
  mem/dcache/req_address/q_reg[21]/CLK           -     CLK
R     (arrival)                   30    57.100          -     4.444
  mem/dcache/req_address/q_reg[21]/Q             -     CLK->Q
F     DFFHQx4_ASAP7_75t_SL        1     57.100     46.800     51.244
  mem/dcache/req_address/FE_PHC3507_FE_RN_3/Y    -     A->Y
F     HB2xp67_ASAP7_75t_SL        1     6.200 20.100     71.344
  mem/dcache/req_address/FE_PHC1862_req_addr_21/Y  -   A->Y
F     HB3xp67_ASAP7_75t_SRAM      3     12.100     83.500   154.844
  mem/dcache/g8402/Y                             -     B->Y
F     XOR2xp5_ASAP7_75t_SL        1     52.100     18.500   173.344
  mem/dcache/g8361/Y                             -     E->Y
F     OR5x1_ASAP7_75t_SL          2     27.800     33.100   206.444
  mem/dcache/g8309/Y                             -     A2->Y
R     O2A1O1Ixp33_ASAP7_75t_SL    2     20.400     21.200   227.644
  mem/dcache/g8307/Y                             -     B->Y
F     NAND2xp5_ASAP7_75t_SL       2     44.500     14.100   241.744
  mem/dcache/g8306/Y                             -     B->Y
R     OAI21xp5_ASAP7_75t_SL       1     26.800     9.900   251.644
  mem/dcache/FE_PHC5254_next_state/Y             -     A->Y
R     HB2xp67_ASAP7_75t_SL        4     22.500     35.900   287.544
  mem/dcache/g8305/Y                             -     B->Y
F     OAI21x1_ASAP7_75t_SL        3     43.700     17.300   304.844
  mem/dcache/FE_OCPC1491_n_256/Y                 -     A->Y
F     HB1xp67_ASAP7_75t_SL        1     30.100     25.800   330.644
  mem/dcache/FE_OCPC1488_n_256/Y                 -     A->Y
R     INVx1_ASAP7_75t_SL          2     24.700     11.300   341.944
  mem/dcache/FE_OFC779_n_256/Y                   -     A->Y
F     INVx1_ASAP7_75t_SL          22    17.200     41.300   383.244
  mem/dcache/g8295/Y                             -     B->Y
R     NAND2xp5_ASAP7_75t_SL       2     79.000     22.500   405.744
  mem/dcache/mem_req_valid                       <<<   mem_req_valid  R
      cache_1                     -          -          -   405.744
  mem/arbiter/g544/Y                             -     B->Y
R     OR2x2_ASAP7_75t_SRAM        1     37.000     47.100   452.844
  FE_OFC363_mem_req_valid/Y                      -     A->Y
R     BUFx24_ASAP7_75t_SL         1     40.300   454.700   907.544
  mem_req_valid                                  <<<   mem_req_valid  R
      -                           1  1369.700   438.700   907.544
```

```
#-------------------------------------------------------------------
-------------------------------------------------------------------
-
```

From this report, the final route from the memory arbiter to the off-chip ExtMemModel delays the signal by nearly the entire clock period. In an older PAR iteration with a target clock of 1ns, this single jump from mem/arbiter to mem_req_valid took 1.2ns.

This delay vastly limits the overall speed of the system. Despite completing a successful synthesis at 0.5ns, and having a PAR tapeout that met timing requirements at 0.6ns, the best simulated clock period is at 1.0ns. The overall delay was slightly reduced by forcing the cache logic and memory arbiter to be as close as possible to the output pins, but in the best case the delay from arbiter to external memory was near or larger than the target clock period.

5) Final Clock Tree:



insertion delay max: 82ps
skew max: 55ps

6) area used

The final design used 69192.202752 um^2.

The majority of the CPU area was from the ID stage (6891um^2), as the register file is spacious and the hazard control logic is complex. The caches took up double the area of the CPU, with 46293.437952um^2 combined.

7) Cycles and Optimizations:

Bmark Short:

```
  [ PASSED ]
/home/cc/eecs151/sp20/class/eecs151-aad/asic_proj/bmark_short_output/cachetest
.out     () after 108252 simulation cycles
```

```
  [ PASSED ]
/home/cc/eecs151/sp20/class/eecs151-aad/asic_proj/bmark_short_output/final.out
() after 9004 simulation cycles
  [ PASSED ]
/home/cc/eecs151/sp20/class/eecs151-aad/asic_proj/bmark_short_output/fib.out
() after 7101 simulation cycles
  [ PASSED ]
/home/cc/eecs151/sp20/class/eecs151-aad/asic_proj/bmark_short_output/sum.out
() after 98190 simulation cycles
  [ PASSED ]
/home/cc/eecs151/sp20/class/eecs151-aad/asic_proj/bmark_short_output/replace.o
ut     () after 98271 simulation cycles
```

ASM:

```
  [ PASSED ]
/home/cc/eecs151/sp20/class/eecs151-aad/asic_proj/asm_output/addi.out     ()
after 393 simulation cycles
  [ PASSED ]
/home/cc/eecs151/sp20/class/eecs151-aad/asic_proj/asm_output/add.out     ()
after 694 simulation cycles
  [ PASSED ]
/home/cc/eecs151/sp20/class/eecs151-aad/asic_proj/asm_output/andi.out     ()
after 331 simulation cycles
  [ PASSED ]
/home/cc/eecs151/sp20/class/eecs151-aad/asic_proj/asm_output/and.out     ()
after 708 simulation cycles
  [ PASSED ]
/home/cc/eecs151/sp20/class/eecs151-aad/asic_proj/asm_output/auipc.out     ()
after 144 simulation cycles
  [ PASSED ]
/home/cc/eecs151/sp20/class/eecs151-aad/asic_proj/asm_output/beq.out     ()
after 488 simulation cycles
  [ PASSED ]
/home/cc/eecs151/sp20/class/eecs151-aad/asic_proj/asm_output/bge.out     ()
after 536 simulation cycles
  [ PASSED ]
/home/cc/eecs151/sp20/class/eecs151-aad/asic_proj/asm_output/bgeu.out     ()
after 561 simulation cycles
  [ PASSED ]
/home/cc/eecs151/sp20/class/eecs151-aad/asic_proj/asm_output/blt.out     ()
after 488 simulation cycles
  [ PASSED ]
/home/cc/eecs151/sp20/class/eecs151-aad/asic_proj/asm_output/bltu.out     ()
after 519 simulation cycles
  [ PASSED ]
/home/cc/eecs151/sp20/class/eecs151-aad/asic_proj/asm_output/bne.out     ()
after 492 simulation cycles
```

```
  [ PASSED ]
/home/cc/eecs151/sp20/class/eecs151-aad/asic_proj/asm_output/jal.out     ()
after 142 simulation cycles
  [ PASSED ]
/home/cc/eecs151/sp20/class/eecs151-aad/asic_proj/asm_output/jalr.out     ()
after 242 simulation cycles
  [ PASSED ]
/home/cc/eecs151/sp20/class/eecs151-aad/asic_proj/asm_output/lb.out     ()
after 399 simulation cycles
  [ PASSED ]
/home/cc/eecs151/sp20/class/eecs151-aad/asic_proj/asm_output/lbu.out     ()
after 399 simulation cycles
  [ PASSED ]
/home/cc/eecs151/sp20/class/eecs151-aad/asic_proj/asm_output/lh.out     ()
after 417 simulation cycles
  [ PASSED ]
/home/cc/eecs151/sp20/class/eecs151-aad/asic_proj/asm_output/lhu.out     ()
after 424 simulation cycles
  [ PASSED ]
/home/cc/eecs151/sp20/class/eecs151-aad/asic_proj/asm_output/lui.out     ()
after 156 simulation cycles
  [ PASSED ]
/home/cc/eecs151/sp20/class/eecs151-aad/asic_proj/asm_output/lw.out     ()
after 427 simulation cycles
  [ PASSED ]
/home/cc/eecs151/sp20/class/eecs151-aad/asic_proj/asm_output/ori.out     ()
after 344 simulation cycles
  [ PASSED ]
/home/cc/eecs151/sp20/class/eecs151-aad/asic_proj/asm_output/or.out     ()
after 711 simulation cycles
  [ PASSED ]
/home/cc/eecs151/sp20/class/eecs151-aad/asic_proj/asm_output/sb.out     ()
after 833 simulation cycles
  [ PASSED ]
/home/cc/eecs151/sp20/class/eecs151-aad/asic_proj/asm_output/sh.out     ()
after 911 simulation cycles
  [ PASSED ]
/home/cc/eecs151/sp20/class/eecs151-aad/asic_proj/asm_output/simple.out     ()
after 118 simulation cycles
  [ PASSED ]
/home/cc/eecs151/sp20/class/eecs151-aad/asic_proj/asm_output/slli.out     ()
after 392 simulation cycles
  [ PASSED ]
/home/cc/eecs151/sp20/class/eecs151-aad/asic_proj/asm_output/sll.out     ()
after 728 simulation cycles
  [ PASSED ]
/home/cc/eecs151/sp20/class/eecs151-aad/asic_proj/asm_output/slti.out     ()
after 388 simulation cycles
```

```
  [ PASSED ]
/home/cc/eecs151/sp20/class/eecs151-aad/asic_proj/asm_output/sltiu.out     ()
after 388 simulation cycles
  [ PASSED ]
/home/cc/eecs151/sp20/class/eecs151-aad/asic_proj/asm_output/slt.out     ()
after 682 simulation cycles
  [ PASSED ]
/home/cc/eecs151/sp20/class/eecs151-aad/asic_proj/asm_output/sltu.out     ()
after 682 simulation cycles
  [ PASSED ]
/home/cc/eecs151/sp20/class/eecs151-aad/asic_proj/asm_output/srai.out     ()
after 413 simulation cycles
  [ PASSED ]
/home/cc/eecs151/sp20/class/eecs151-aad/asic_proj/asm_output/sra.out     ()
after 759 simulation cycles
  [ PASSED ]
/home/cc/eecs151/sp20/class/eecs151-aad/asic_proj/asm_output/srli.out     ()
after 407 simulation cycles
  [ PASSED ]
/home/cc/eecs151/sp20/class/eecs151-aad/asic_proj/asm_output/srl.out     ()
after 747 simulation cycles
  [ PASSED ]
/home/cc/eecs151/sp20/class/eecs151-aad/asic_proj/asm_output/sub.out     ()
after 680 simulation cycles
  [ PASSED ]
/home/cc/eecs151/sp20/class/eecs151-aad/asic_proj/asm_output/sw.out     ()
after 911 simulation cycles
  [ PASSED ]
/home/cc/eecs151/sp20/class/eecs151-aad/asic_proj/asm_output/xori.out     ()
after 346 simulation cycles
  [ PASSED ]
/home/cc/eecs151/sp20/class/eecs151-aad/asic_proj/asm_output/xor.out     ()
after 710 simulation cycles
```

The major optimization made was a more aggressive pipeline to boost the clock speed. Since the pipeline already had a stage between the register file / hazard logic and the ALU, the only needed change was splitting up the instruction cache and the register file, and adding registers between the dcache and the writeback mux. One optimization issue was the writeback mux had to be replicated at the end of the MEM stage to provide outputs to the bypass logic in the correct stage, but since this wasn't the critical path it did not hinder performance. Initially, the five-stage pipeline had a combined write-back / ID stage with double-pumping on the register file. Since the bypassing logic was so complex, it was easier just to put the writeback mux in the MEM stage. In theory, the commit point of the processor is the fifth stage:
Instruction fetch (icache) | Instruction decode (rf read and hazard / alu logic) | execute (ALU) | MEM (dcache and writeback mux) | commit (register file has written, CSR is written to).

Adding the extra stages improved clock speed, but the CPI went down. In the initial system, JAL commands were resolved in one cycle (no bubbles needed) and dynamic branches and jumps were resolved in the second cycle (one bubble inserted). After splitting up IF and ID, one bubble must be inserted for a JAL command, and two bubbles must be inserted after a taken branch or a JALR command.

This optimization was not the most useful in the end, as the speed of the system was limited by the off-chip delay rather than the logic itself. At one point, we had a post-par report with no timing violations at a 6ns clock speed. If this interfaced with memory in a reasonable time, the overall speedup would be much greater. If we had more time, or discovered the cause of this limitation sooner, a potential fix would be to add a register between the memory arbiter and the output pin, and simply add a cycle of delay to external memory requests.

With additional time, it is likely the system would benefit from a 2-bit branch predictor. The delay of a small (64 entry) branch predictor with a tag, cached destination and two bits designating the branch prediction would be lower than that of the cache, so it wouldn't contribute to the critical path. Similarly, implementing a return stack that detects JALR x0 0(x2) would be able to speed up return calls to result in one or zero bubbles.

division of labor:
labor was pretty evenly distributed for the most part