COMP. ARCH. II:

- ASSUMED SINGLE CYCLE OPERATION SO FAR

- DRAWBACK: MUST SET CLOCK FREQ FOR SLOWEST OPERATION!

- E.g. IF MEM ACCESS (LD) IS ~~200ps~~ 200 ps AND REG ACCESS IS 30 ps, ALU EXECUTION IS 100 ps, IFETCH IS 200 ps

	IFETCH	REG	ALU	DATA	REGS	TOTAL
BRANCH:	200 ps	30 ps	100 ps	0	0	330 ps
LOAD	200 ps	30 ps	100 ps	200 ps	30 ps	560 ps

WORST CASE!

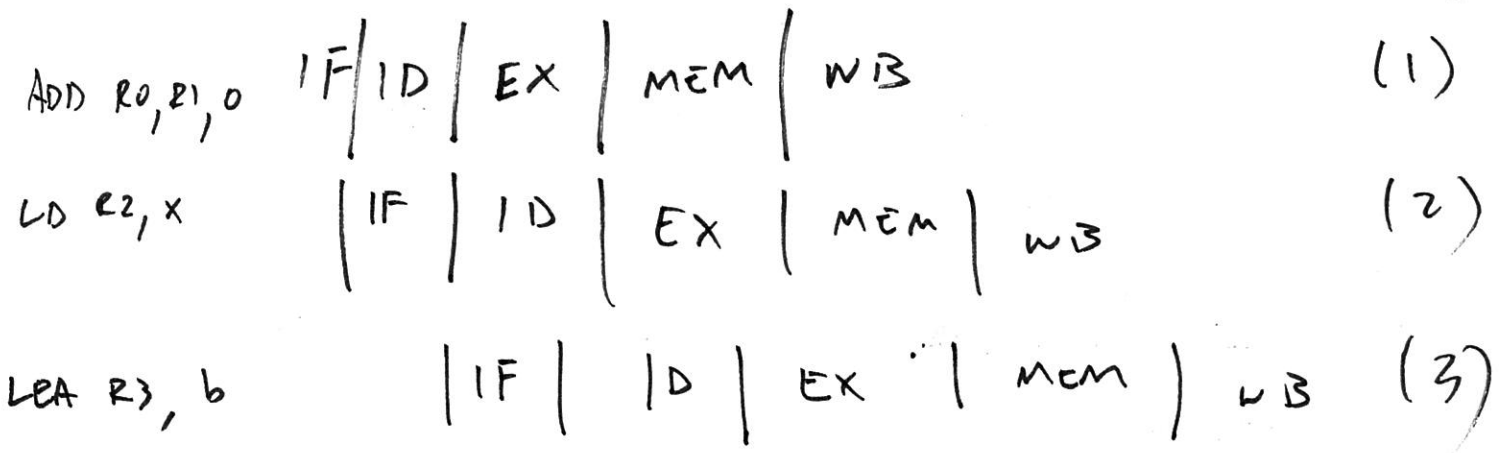
MUST CLOCK AT ~1.8 GHz

- BUT OUR LONGEST PHASE IS ONLY 200 ps!

- IN ~~THEORY~~ THIS SHOULD ALLOW US TO CLOCK AT ~5 GHz

- ALSO, SINCE EVERYTHING OCCURS DURING ONE CLOCK CYCLE, WE MUST REPLICATE FUNCTIONAL UNITS! (E.g. ADDERS FOR ALU & PC CALCULATION)

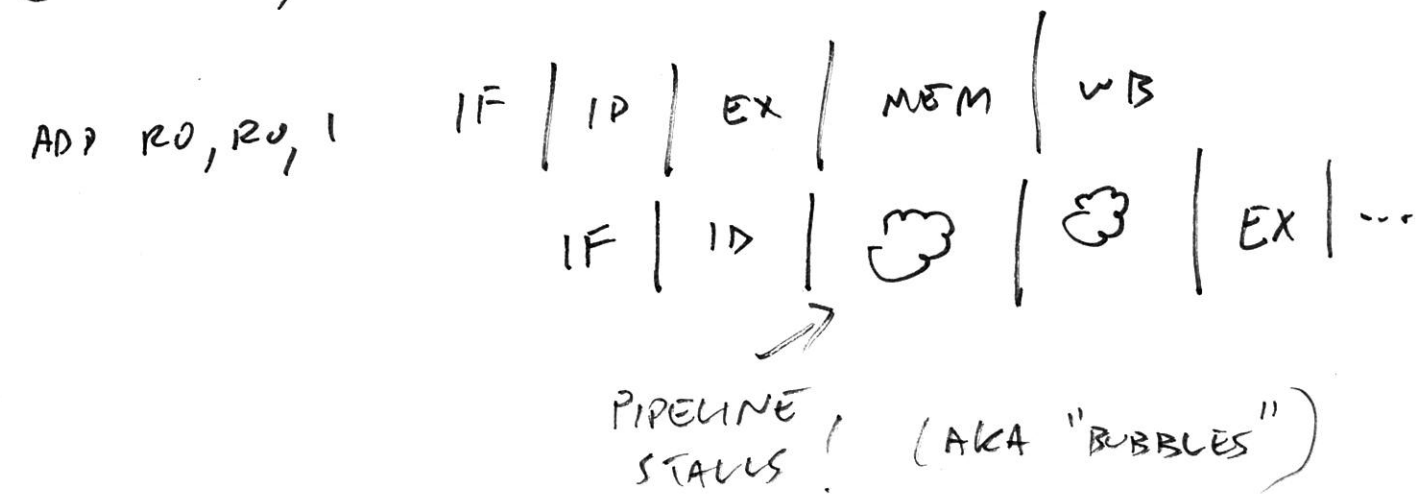
- SOLUTION? PIPELINING (CLASSIC 5-STAGE EXAMPLE)



- THIS INCREASES THROUGHPUT OF THE PROCESSOR. LATENCY PER INSTRUCTION MAY ACTUALLY GO UP!

- ~~#~~ #INSTRS/sec ↑
- CAN NOW REUSE FUNCTIONAL UNITS
- WHAT HAPPENS WHEN WE HAVE DEPENDENCIES?

```
{ ADD R0, R0, 1
{ ADD R3, R0, 2
```



- WHAT WE JUST SAW IS A DATA HAZARD (LATER INSTRUCTION DEPENDS ON EARLIER ONE SO WE MUST WAIT)

- 1) SCHEDULE INSTRUCTIONS DIFFERENTLY (THROW A NON-DEPENDENT INSTRUCTION IN-BETWEEN)

  * COMPILER DOES IT: STATIC SCHEDULING
  * CPU DOES IT: DYNAMIC SCHEDULING

IF INSTRUCTIONS ARE EXECUTED OUT OF PROGRAM ORDER, WE CALL IT AN OUT-OF-ORDER PROCESSOR.

A LOT OF SPECIAL LOGIC NEEDED FOR THIS! EXPENSIVE

- 2) [redacted]

FORWARD RESULTS! ALSO CALLED BYPASSING

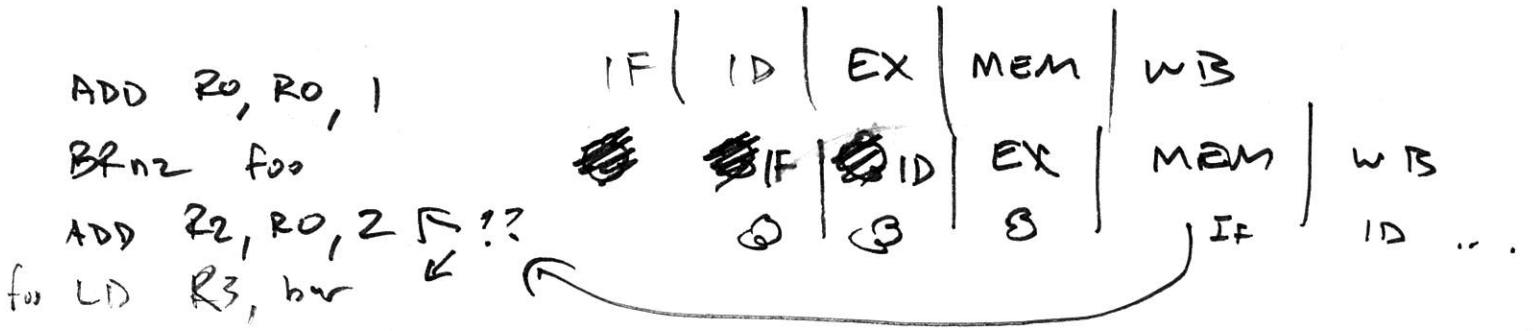
BASIC IDEA: WHY WAIT TO WRITE BACK RESULTS WHEN THEY'RE IN THE PIPE?

- WE CAN ALSO HAVE STRUCTURAL HAZARDS:

MORE THAN 1 INSTR. TRYING TO USE SAME RESOURCE.

E.g. double-pump REGISTER FILE, OR ALLOW CONCURRENT READS/WRITES ("MULTI-PORTED" REG FILE)

- CONTROL HAZARDS: INSTRUCTION EXEC. DEPENDS ON PREVIOUS RESULTS — BRANCHES!



JOIN:

BRANCH PREDICTION!

GUESS! EASIEST? ALWAYS GUESS NOT TAKEN

THIS CAN ACTUALLY WORK...

```
        LD  R0, COUNT
Loop    ADD R0, R0, -1
        BRnz End
        BR  Loop
End     HALT
```

HERE, WE'LL ONLY MISPREDICT $\frac{1}{\text{COUNT}}$ OF THE TIME

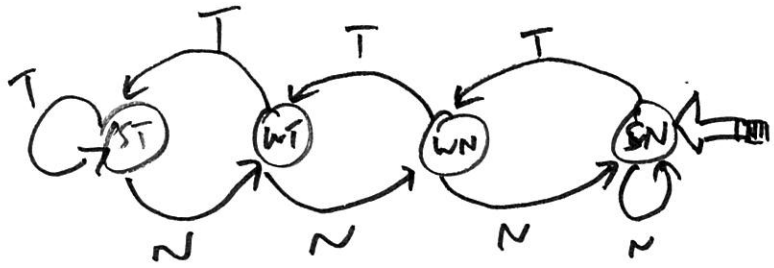
MORE SOPHISTICATED: USE HISTORY

- INDEX BY PC OF BRANCH (LOW BITS)

LAST TAKEN? TAKE

NOT? DON'T

NEXT STEP: TWO BITS ...



ST: "STRONGLY TAKEN"
WT: "WEAKLY TAKEN"
WN: "WEAKLY NOT TAKEN"
SN: "STRONGLY NOT TAKEN"

2-BIT SATURATING COUNTER

- CAN DIFFERENTIATE BETWEEN ALIASED BRANCHES (LOW ORDER BITS OF PC ARE SAME)
  - JUST LIKE CACHES, ADD A TAG
- ADD MORE HISTORY