What was it like….

- 1983:
  - Had CMOS and VLSI revolution
  - 64 – 512k
  - There was RISC (Sun)
  - There was C

What is a VLIW machine?

- Explicitly Parallel Machine
  - 1 thread of control
  - Bundle of statically scheduled instructions

Trace Scheduling

- Optimization of most likely path
  - Step 1: Choose the likely path
    - Compiler hints
    - Profile-guided optimization
    - Likely statistics
  - Step 2: Group multiple basic blocks together and remove branches
  - Step 3: Decipher all the bad conditions and make fixup blocks
- Gives us: A large basic block

Trace Scheduling – How?

Branching

- Average 6-7 instructions per basic block (normal)
- [i0][i1][i2][i3][b1][i4][b2][i4]

Difficulties & Solutions

- What if they all access the same memory address?

- Resource conflict on functional unit
  - Restrict generality
- Dataflow dependencies
  - Large bank of registers so that adjacent instructions don't use the same
  - Restrict generality

Memory bandwidth

Memory Disambiguation – pointers?

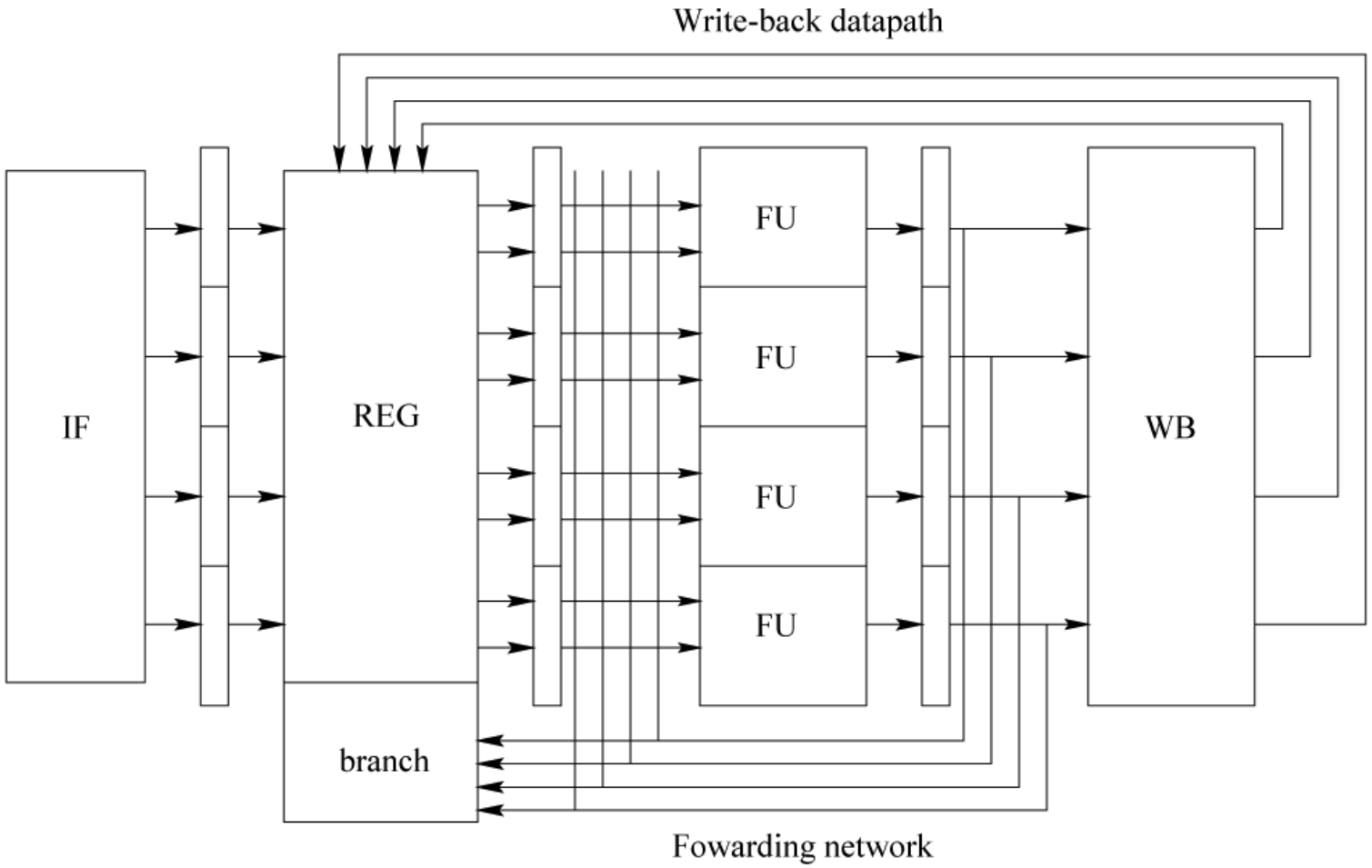
- Scientific codes
  - Fortran

Procedures?

Vector vs. VLIW

- Vector are SIMD and VLIW is MIMD
- VLIW is more genereal
- Can't count on VLIW performance, vector is highly predictable
- Vector is less general => can be smaller hardware

What information is only available dynamically?


Write-back datapath
IF
REG
branch
FU
FU
FU
FU
WB
Fowarding network

Software pipelining

- For(x=0;x<j;x++) {
  r[x] = a[x]+b[x];
  if(r[x]>255)
  r[x]=255;
  }