Where are the Data Dependencies? Where are the Stalls?

```
Loop:   LD      F0, 0(R1)       ;F0=vector element
        ADDD    F4, F0, F2      ;add scalar from F2
        SD      0(R1), F4       ;store result
        SUBI    R1, R1, 8       ;decrement pointer 8B (DW)
        BNEZ    R1, Loop        ;branch R1 != 0
        NOP                     ;delayed branch slot
```

Instruction Producing Result	*Instruction Using Result*	*Latency in Clock Cycles*
FP ALU Op	Another FP ALU Op	3
FP ALU Op	Store Double	2
Load Double	FP ALU Op	1
Load Double	Store Double	0
Integer Op	Integer Op	0

Revised FP Loop Minimizing Stalls

```
Loop:   LD      F0, 0(R1
        <stall>
        ADDD    F4, F0, F2
        SUBI    R1, R1, 8
        BNEZ    R1, Loop        ;delayed branch
        SD      8(R1), F4       ;altered when moved past SUBI
```

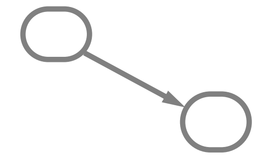
- NOTE: Swap BNEZ and SD by changing address of SD
- 6 Cycles Per Element
- Next step: Unroll loop

Instruction Producing Result	*Instruction Using Result*	*Latency in Clock Cycles*
FP ALU Op	Another FP ALU Op	3
FP ALU Op	Store Double	2
Load Double	FP ALU Op	1
Load Double	Store Double	0
Integer Op	Integer Op	0

Naive Unrolled Loop

```
Loop: LD    F0, 0(R1)
      ADDD  F4, F0, F2       stall
      SD    0(R1), F4        2 stalls   ;drop SUBI & BNEZ
      LD    F0, -8(R1)
      ADDD  F4, F0, F2
      SD    -8(R1), F4                  ;drop SUBI & BNEZ
      LD    F0, -16(R1)
      ADDD  F4, F0, F2
      SD    -16(R1), F4                 ;drop SUBI & BNEZ
      LD    F0, -24(R1)
      ADDD  F4, F0, F2
      SD    -24(R1), F4
      SUBI  R1, R1, #32                 ;alter to 4*8
      BNEZ  R1, Loop
      NOP
```

True Dependencies Marked



- Got rid of "fluff"
- Data dependencies --> Hazards (6 cycles per element + 3 loop overhead)
- Dependencies limit reordering

  Data dependencies --> partial order

  Data + name dependencies --> nearly total order
- Solution: Compiler renaming

Unrolled, Scheduled Loop

```
Loop:   LD      F0, 0(R1)
        LD      F6, -8(R1)
        LD      F10, -16(R1)
        LD      F14, -24(R1)
        ADDD    F4, F0, F2
        ADDD    F8, F6, F2
        ADDD    F12, F10, F2
        ADDD    F16, F14, F2
        SD      0(R1), F4
        SD      -8(R1), F8
        SD      -16(R1), F12
        SUBI    R1, R1, #32
        BNEZ    R1, Loop
        SD      8(R1), F16          ; 8-32 = -24
```

- Assumes R1 is a multiple of 4 (Prolog code)
- 14 Clock Cycles = 3.5 per iteration
- Uses compiler register renaming

Where are the control dependencies?

```
Loop:   LD      F0, 0(R1)
        ADDD    F4, F0, F2
        SD      0(R1), F4
        SUBI    R1, R1, 8
        BEQZ    R1, exit
        LD      F0, 0(R1)
        ADDD    F4, F0, F2
        SD      0(R1), F4
        SUBI    R1, R1, 8
        BEQZ    R1, exit
        LD      F0, 0(R1)
        ADDD    F4, F0, F2
        SD      0(R1), F4
        SUBI    R1, R1, 8
        BEQZ    R1, exit
        ...
```

When Safe to Unroll Loop?

Where are data dependencies
(A, B, C distinct and nonoverlapping)

```
A[i+1]    = A[i] + C[i];       // S1
B[i+1]    = B[i] + A[i+1];     // S2
```