Program Behavior (Continued)

Though there is no one answer to the question "How does a program behave?" it is useful to look carefully at the properties of programs.

Breaking a pipeline -- encountering a situation where an instruction execution cannot be continued in the next clock tick -- is inevitable. Reducing such situations is an important goal. Examples --

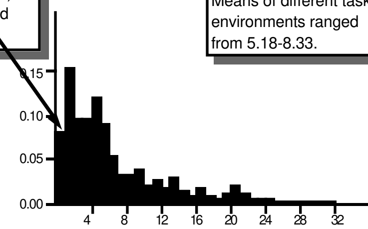
- Branches
- Data dependencies
- Run-on instructions

Instruction Run Length

How many instructions are typically executed before a branch?

Length 1 sequence (8%) = branch taken followed by a branch taken.

Means of different task environments ranged from 5.18-8.33.



Notice: The mean distance between branches is 7.5 instructions, but 66% of the sequences are shorter than that.

Branch Classes

Branches come in a variety of forms. Their usage varies across different applications areas.

Branch Class Occurrence			
Type 1: BC, BR, BCR, BRR	72.5%		
Type 2: Loop Control (BCT...	9.8% (Goto (91%), Fall (9%))		
Type 3: Procedure Call/Return	17.7% (BAL (80%), BALR(20%))		
Type 1 Addressing			
Target in register	17.4%		
Target is effective address	82.6%		
Type 1 Targets	Scientific	Commercial	Systems
Unconditional	20%	40%	35%
Conditional (taken)	43.2%	24.3%	32.5%
Conditional (fall thru)	35.7%	35.7%	32.5%

Computing Properties of Branches

How often does a branch go to its target:

freq of br type × freq br taken = overall freq

Type 1 (BR BC)	$.725\ (0.2 + .432) +$
Type 2 (Loops)	$.098 \times .91 +$
Type 3 (Procs)	$.177 \times 1 = 0.754$

How what fraction of branches use EA?

freq of br type × frac using EA = overall

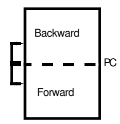
Type 1 (BR BC)	$.725 \times .826 +$
Type 2 (Loops)	$.098 \times 1 +$
Type 3 (Procs)	$.177 \times .8 = 0.729$

Distance from CC Set to Branch	Gen	Sci
Pgm w/max	1.58	1.58
Pgm w/min	0.05	0.95
Geom. mean	0.33	1.24

Fetching Branch Instructions

Foward Prefetch		
Size (bytes)	Backw'd Branch	Forw'd Branch
16	1	32
32	5	32
64	11	35
128	16	39
256	19	42
512	22	46
1024	24	49
2048	26	53
4096	28	55
8192	30	57
16K	33	60
32K	36	63
Inf	36.5	63.5

Centered Table		
Size (bytes)	Config Fwd/Back	Percent
16	±8	13%
32	±16	33%
64	±32	37%
128	±64	46%
256	±128	55%
512	±256	61%
1024	±512	68%
2048	±1024	73%
4096	±2048	79%
8192	±4096	83%
16K	±8192	87%
32K	±16K	93%
64K	±32K	99%

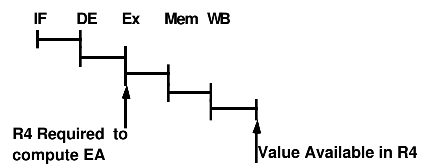


Data Dependencies

Address calculations and computations cannot proceed unless the values they depend on are available.

Computing the Effective Address

```
LD.W    R4,  ...                  LD.W    R4,  ...
SUB.W   R6, Disp[R5, R4]          LD.W    R6, Disp[R5, R4]
                                  SUB.W   R6,R4,R6
```



Data Dependence Stats

Information gathered by hardware monitoring, but without software to reorganize the dependences.

Address Interlock Distance		
%	Cum.%	Dist
9.901	9.901	1 (prev.)
8.397	18.299	2
2.801	21.100	3
2.201	23.301	4
0.258	23.560	5
0.661	24.221	6
0.046	24.268	7
0.360	24.628	8
. . .		

Execution Interlock Distance		
%	Cum.%	Dist
40.297	40.297	1 (prev.)
14.700	54.998	2
3.559	58.557	3
4,898	63.456	4
6.650	70.107	5
1.706	71.813	6
3.217	75.030	7
2.518	77.549	8
. . .		

A 3 instruction addressing interlock costs ...

3~ $\times$ 0.099
2~ $\times$ 0.084
1~ $\times$ 0.028

0.493 cycles for all R/M instructions (.6)

æ Addressing interlock penalty = 0.493 $\times$ 0.6 = 0.3

Run-on Instructions

Most instructions are basic and execute in the time of the execution step in the pipeline.

A few instructions require extremely long time intervals to execute. E.g. Divide, square root.

Run-on instructions stall the pipeline, assuming that instructions are executed in sequence.

The affect on the CPI is significant: A 33 cycle instruction used 2% of the time adds 33×0.02 = 0.66 to the CPI.