Design of Experiments

CS 700

Design of Experiments

- Goals
- Terminology
- Full factorial designs
    - *m*-factor ANOVA
- Fractional factorial designs
- Multi-factorial designs

Recall: One-Factor ANOVA

- Separates total variation observed in a set of measurements into:
  1. Variation within one system
     - Due to random measurement errors
  2. Variation between systems
     - Due to real differences + random error
- Is variation(2) statistically > variation(1)?
- *One-factor experimental design*

ANOVA Summary

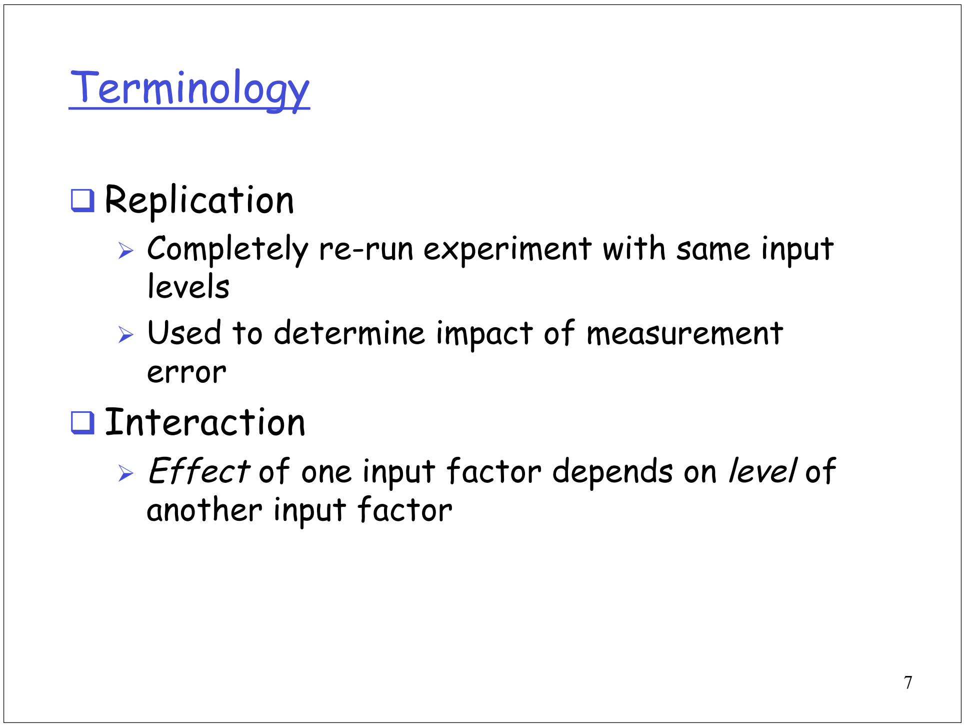
Variation	Alternatives	Error	Total
Sum of squares	SSA	SSE	SST
Deg freedom	$k-1$	$k(n-1)$	$kn-1$
Mean square	$s_a^2 = SSA/(k-1)$	$s_e^2 = SSE/[k(n-1)]$	
Computed F	s_a^2/s_e^2		
Tabulated F	$F_{[1-\alpha;(k-1),k(n-1)]}$		

Generalized Design of Experiments

- Goals
  - Isolate effects of each input variable.
  - Determine effects of interactions.
  - Determine magnitude of experimental error
  - Obtain maximum information for given effort
- Basic idea
  - Expand 1-factor ANOVA to *m* factors

Terminology

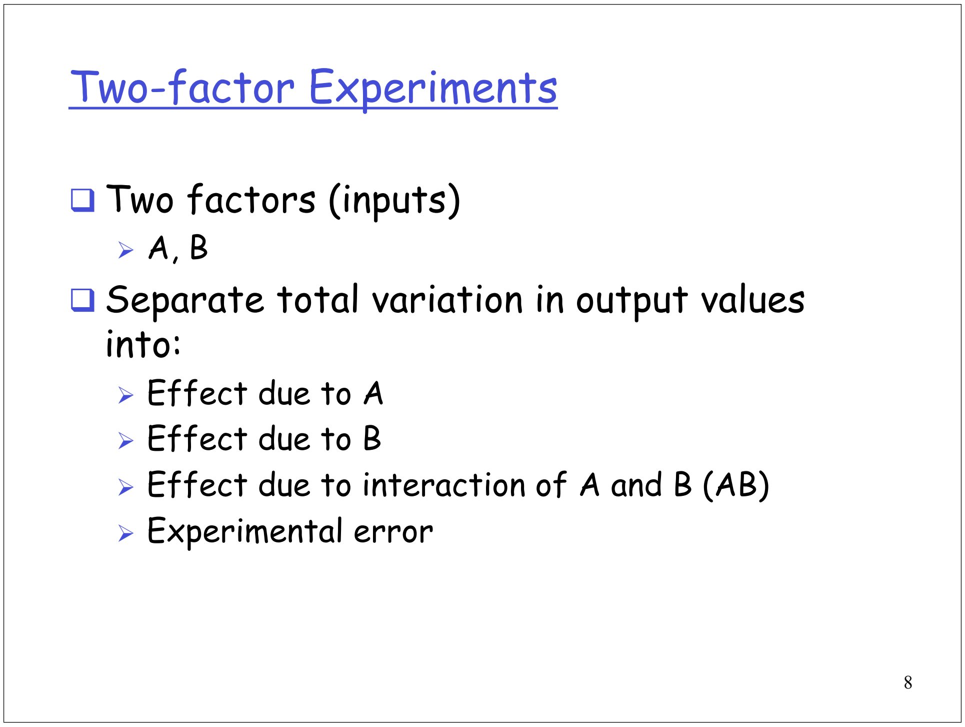
- Response variable
  - Measured output value
    - E.g. total execution time
- Factors
  - Input variables that can be changed
    - E.g. cache size, clock rate, bytes transmitted
- Levels
  - Specific values of factors (inputs)
    - Continuous (~bytes) or discrete (type of system)



Terminology

- ❑ Replication
  - ➢ Completely re-run experiment with same input levels
  - ➢ Used to determine impact of measurement error
- ❑ Interaction
  - ➢ *Effect* of one input factor depends on *level* of another input factor

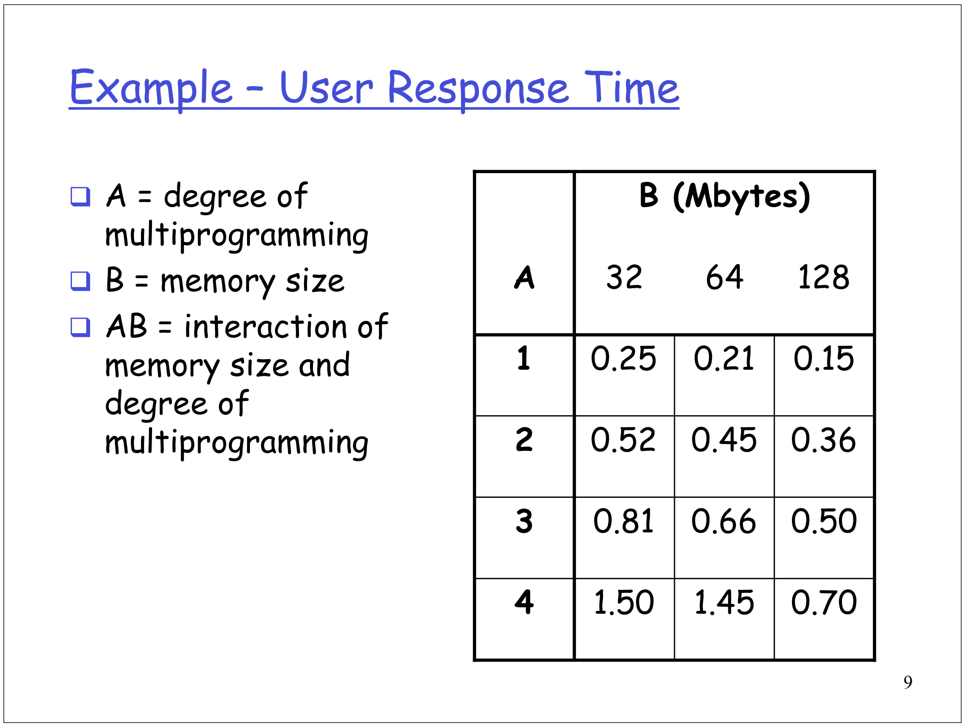
7



Two-factor Experiments

- ❑ Two factors (inputs)
  - ➢ A, B
- ❑ Separate total variation in output values into:
  - ➢ Effect due to A
  - ➢ Effect due to B
  - ➢ Effect due to interaction of A and B (AB)
  - ➢ Experimental error

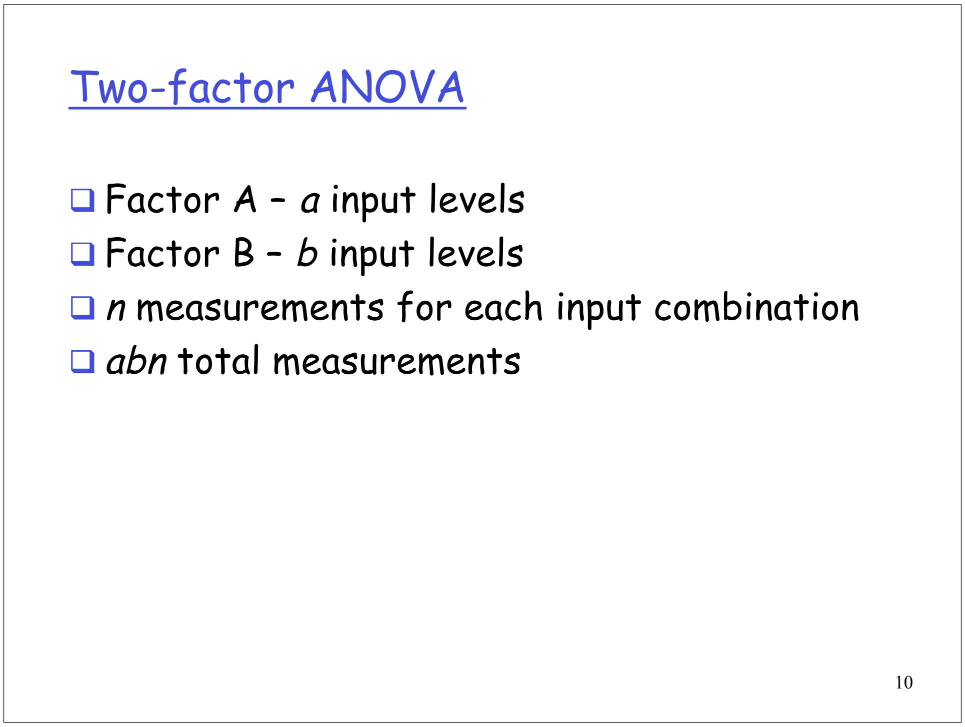
8



Example – User Response Time

- A = degree of multiprogramming
- B = memory size
- AB = interaction of memory size and degree of multiprogramming

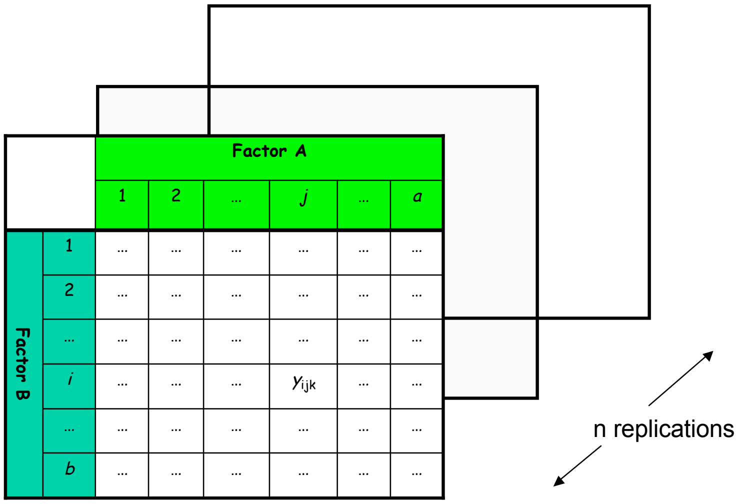
	B (Mbytes)		
A	32	64	128
1	0.25	0.21	0.15
2	0.52	0.45	0.36
3	0.81	0.66	0.50
4	1.50	1.45	0.70



Two-factor ANOVA

- Factor A – *a* input levels
- Factor B – *b* input levels
- *n* measurements for each input combination
- *abn* total measurements

Two Factors, *n* Replications



		Factor A					
		1	2	…	*j*	…	*a*
Factor B	1	…	…	…	…	…	…
	2	…	…	…	…	…	…
	…	…	…	…	…	…	…
	i	…	…	…	y_{ijk}	…	…
	…	…	…	…	…	…	…
	b	…	…	…	…	…	…

Recall: One-factor ANOVA

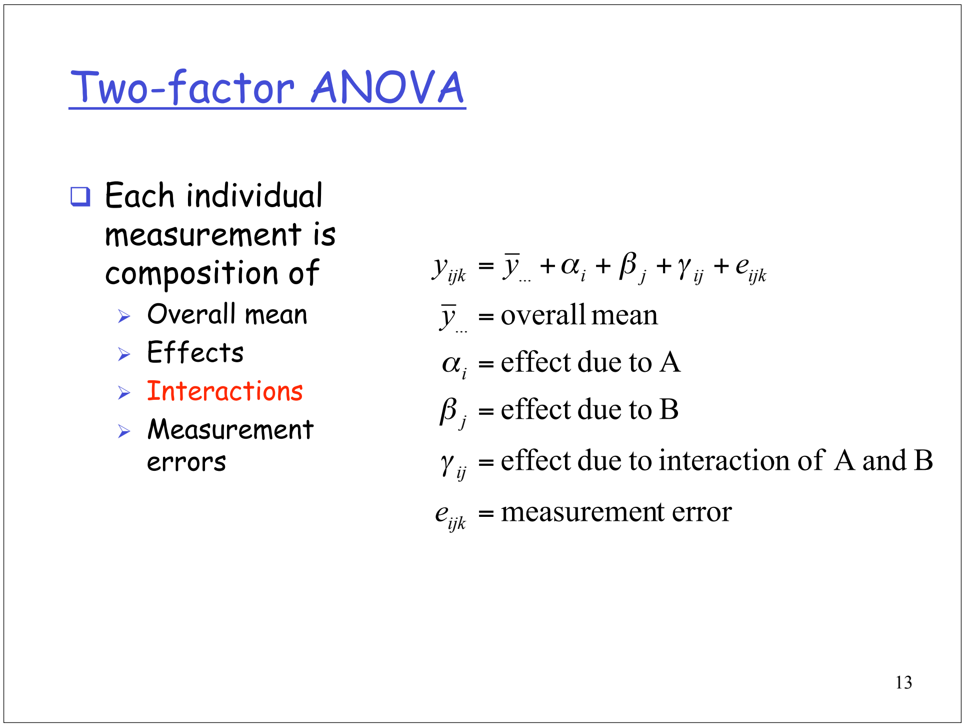
- Each individual measurement is composition of
  - Overall mean
  - Effect of alternatives
  - Measurement errors

$$y_{ij} = \bar{y}_{..} + \alpha_i + e_{ij}$$

$$\bar{y}_{..} = \text{overall mean}$$

$$\alpha_i = \text{effect due to A}$$

$$e_{ij} = \text{measurement error}$$



Two-factor ANOVA

- ❑ Each individual measurement is composition of
  - Overall mean
  - Effects
  - Interactions
  - Measurement errors

$$y_{ijk} = \bar{y}_{...} + \alpha_i + \beta_j + \gamma_{ij} + e_{ijk}$$

$\bar{y}_{...}$ = overall mean

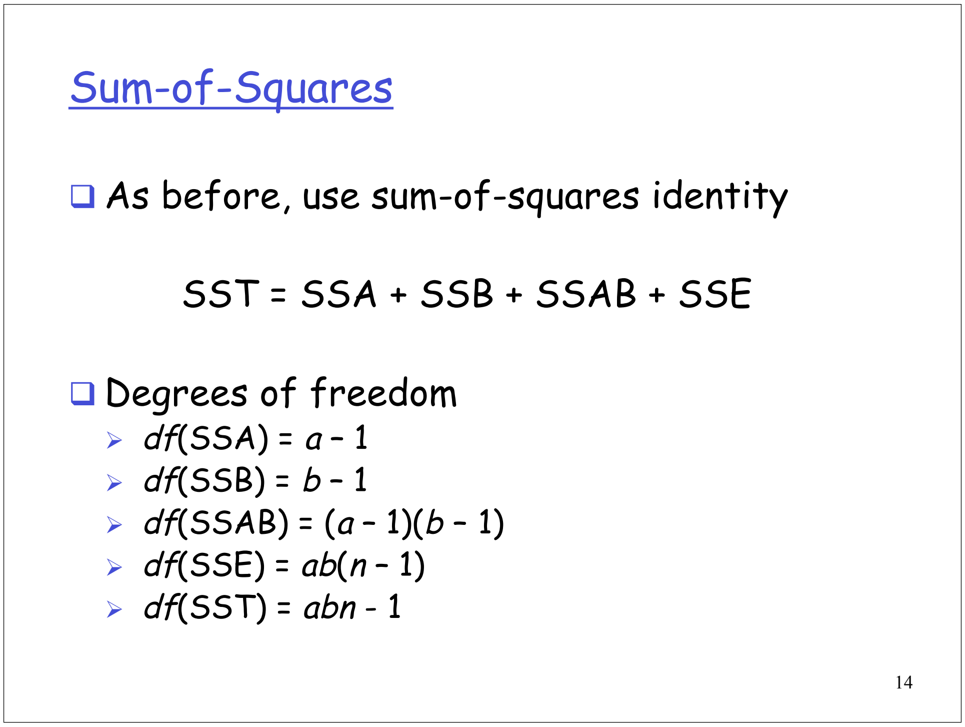
α_i = effect due to A

β_j = effect due to B

γ_{ij} = effect due to interaction of A and B

e_{ijk} = measurement error

13



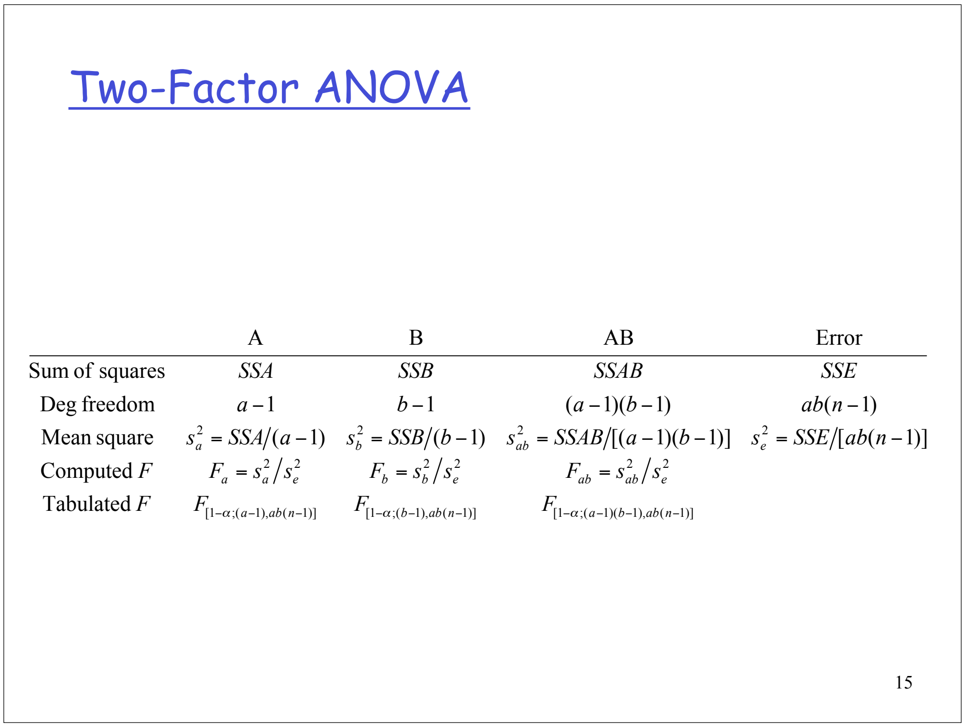
Sum-of-Squares

- ❑ As before, use sum-of-squares identity

SST = SSA + SSB + SSAB + SSE

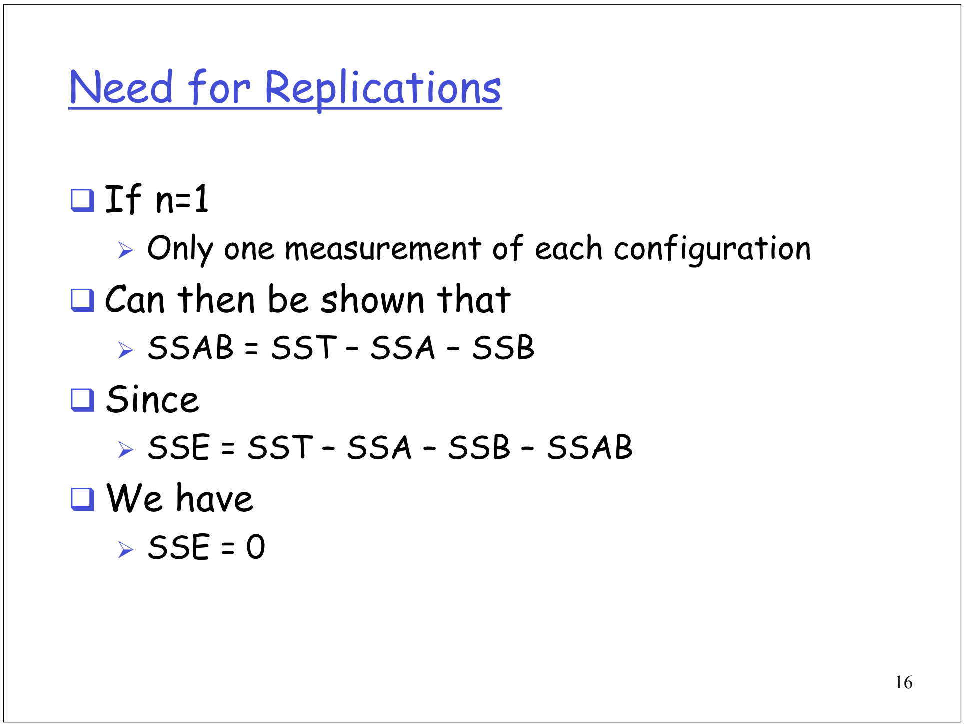
- ❑ Degrees of freedom
  - *df*(SSA) = *a* – 1
  - *df*(SSB) = *b* – 1
  - *df*(SSAB) = (*a* – 1)(*b* – 1)
  - *df*(SSE) = *ab*(*n* – 1)
  - *df*(SST) = *abn* - 1

14



Two-Factor ANOVA

	A	B	AB	Error
Sum of squares	SSA	SSB	$SSAB$	SSE
Deg freedom	$a-1$	$b-1$	$(a-1)(b-1)$	$ab(n-1)$
Mean square	$s_a^2 = SSA/(a-1)$	$s_b^2 = SSB/(b-1)$	$s_{ab}^2 = SSAB/[(a-1)(b-1)]$	$s_e^2 = SSE/[ab(n-1)]$
Computed F	$F_a = s_a^2/s_e^2$	$F_b = s_b^2/s_e^2$	$F_{ab} = s_{ab}^2/s_e^2$	
Tabulated F	$F_{[1-\alpha;(a-1),ab(n-1)]}$	$F_{[1-\alpha;(b-1),ab(n-1)]}$	$F_{[1-\alpha;(a-1)(b-1),ab(n-1)]}$	



Need for Replications

- ❑ If n=1
  - ➢ Only one measurement of each configuration
- ❑ Can then be shown that
  - ➢ SSAB = SST - SSA - SSB
- ❑ Since
  - ➢ SSE = SST - SSA - SSB - SSAB
- ❑ We have
  - ➢ SSE = 0

Need for Replications

- Thus, when n=1
  - SSE = 0
  - → No information about measurement errors
- Cannot separate effect due to interactions from measurement noise
- Must *replicate* each experiment at least twice

Example

- Output = user response time (seconds)
- Want to separate effects due to
  - A = degree of multiprogramming
  - B = memory size
  - AB = interaction
  - Error
- Need replications to separate error

	B (Mbytes)		
A	32	64	128
1	0.25	0.21	0.15
2	0.52	0.45	0.36
3	0.81	0.66	0.50
4	1.50	1.45	0.70

Example

	B (Mbytes)		
A	32	64	128
1	0.25	0.21	0.15
	0.28	0.19	0.11
2	0.52	0.45	0.36
	0.48	0.49	0.30
3	0.81	0.66	0.50
	0.76	0.59	0.61
4	1.50	1.45	0.70
	1.61	1.32	0.68

Example

	A	B	AB	Error
Sum of squares	3.3714	0.5152	0.4317	0.0293
Deg freedom	3	2	6	12
Mean square	1.1238	0.2576	0.0720	0.0024
Computed F	460.2	105.5	29.5	
Tabulated F	$F_{[0.95;3,12]} = 3.49$	$F_{[0.95;2,12]} = 3.89$	$F_{[0.95;6,12]} = 3.00$	

Conclusions From the Example

- 77.6% (SSA/SST) of all variation in response time due to degree of multiprogramming
- 11.8% (SSB/SST) due to memory size
- 9.9% (SSAB/SST) due to interaction
- 0.7% due to measurement error
- 95% confident that all effects and interactions are statistically significant

Generalized *m*-factor Experiments

m factors $\Rightarrow$

m main effects

$\binom{m}{2}$ two - factor interactions

$\binom{m}{3}$ three - factor interactions

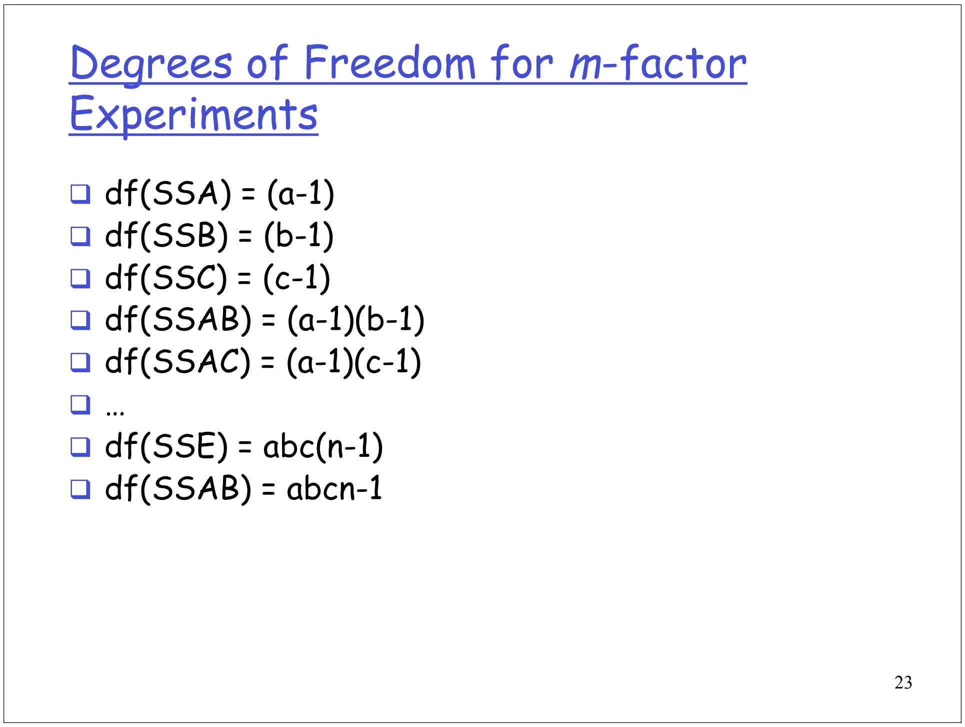
$\vdots$

$\binom{m}{m} = 1$ m - factor interactions

$2^m - 1$ total effects

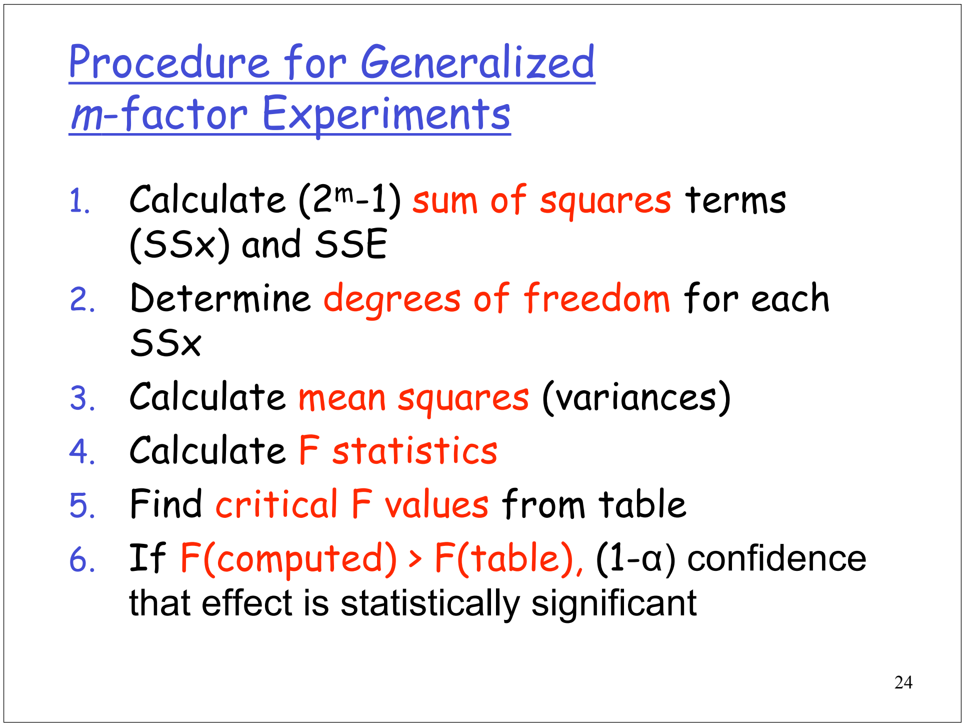
Effects for 3 factors:

A
B
C
AB
AC
BC
ABC



Degrees of Freedom for *m*-factor Experiments

- df(SSA) = (a-1)
- df(SSB) = (b-1)
- df(SSC) = (c-1)
- df(SSAB) = (a-1)(b-1)
- df(SSAC) = (a-1)(c-1)
- …
- df(SSE) = abc(n-1)
- df(SSAB) = abcn-1



Procedure for Generalized *m*-factor Experiments

1. Calculate (2^m-1) sum of squares terms (SSx) and SSE
2. Determine degrees of freedom for each SSx
3. Calculate mean squares (variances)
4. Calculate F statistics
5. Find critical F values from table
6. If F(computed) > F(table), (1-α) confidence that effect is statistically significant

A Problem

- ❑ *Full factorial design with replication*
  - ➢ Measure system response with all possible input combinations
  - ➢ Replicate each measurement *n* times to determine effect of measurement error
- ❑ *m* factors, *v* levels, *n* replications

  → $n\ v^m$ experiments
- ❑ m = 5 input factors, v = 4 levels, n = 3
  - ➢ → $3(4^5)$ = 3,072 experiments!

Fractional Factorial Designs: $n2^m$ Experiments

- ❑ Special case of generalized *m*-factor experiments
- ❑ Restrict each factor to two possible values
  - ➢ High, low
  - ➢ On, off
- ❑ Find factors that have largest impact
- ❑ Full factorial design with only those factors

$n2^m$ Experiments

	A	B	AB	Error
Sum of squares	SSA	SSB	$SSAB$	SSE
Deg freedom	1	1	1	$2^m(n-1)$
Mean square	$s_a^2 = SSA/1$	$s_b^2 = SSB/1$	$s_{ab}^2 = SSAB/1$	$s_e^2 = SSE/[2^m(n-1)]$
Computed F	$F_a = s_a^2/s_e^2$	$F_b = s_b^2/s_e^2$	$F_{ab} = s_{ab}^2/s_e^2$	
Tabulated F	$F_{[1-\alpha;1,2^m(n-1)]}$	$F_{[1-\alpha;1,2^m(n-1)]}$	$F_{[1-\alpha;1,2^m(n-1)]}$	

Finding Sum of Squares Terms

Sum of *n* measurements with (A,B) = (High, Low)	Factor A	Factor B
y_{AB}	High	High
y_{Ab}	High	Low
y_{aB}	Low	High
y_{ab}	Low	Low

$n2^m$ Contrasts

$$w_A = y_{AB} + y_{Ab} - y_{aB} - y_{ab}$$

$$w_B = y_{AB} - y_{Ab} + y_{aB} - y_{ab}$$

$$w_{AB} = y_{AB} - y_{Ab} - y_{aB} + y_{ab}$$

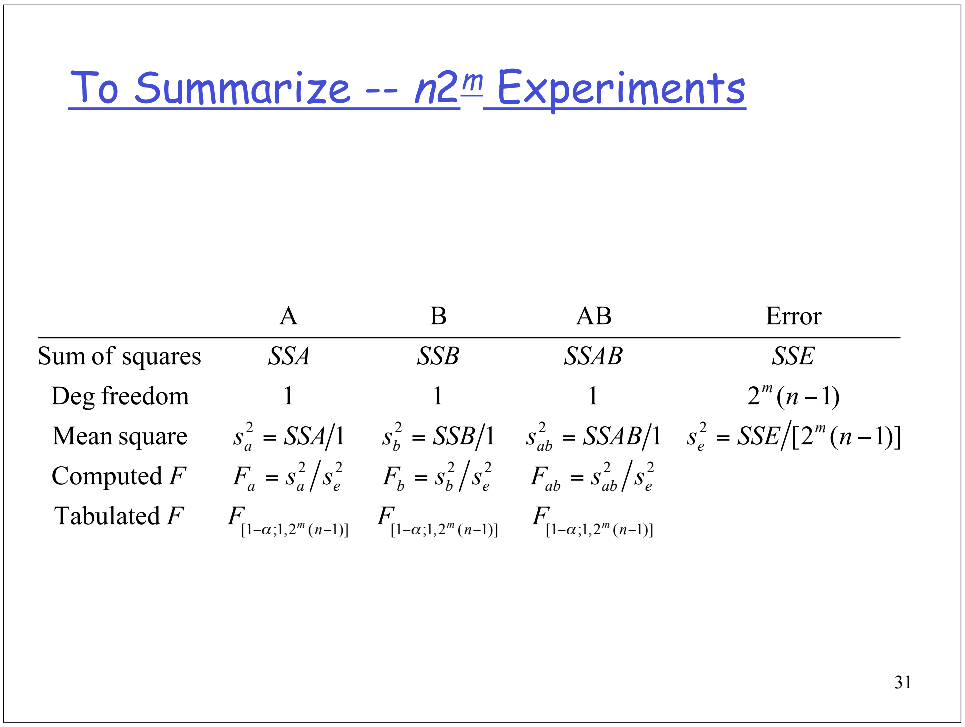
$n2^m$ Sum of Squares

$$SSA = \frac{w_A^2}{n2^m}$$

$$SSB = \frac{w_B^2}{n2^m}$$

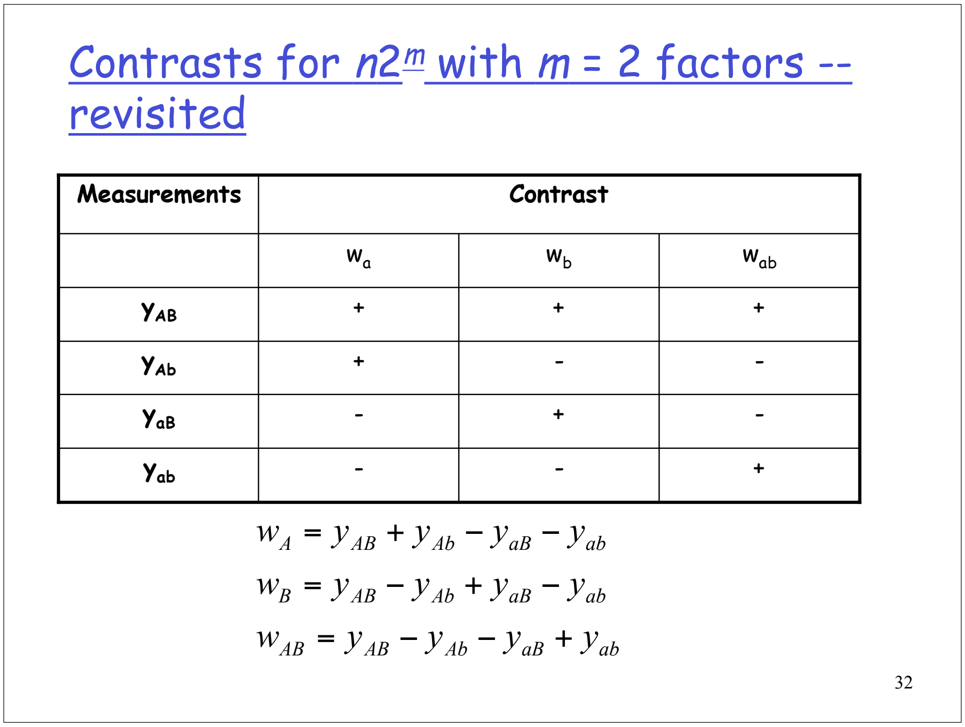
$$SSAB = \frac{w_{AB}^2}{n2^m}$$

$$SSE = SST - SSA - SSB - SSAB$$



To Summarize -- $n2^m$ Experiments

	A	B	AB	Error
Sum of squares	SSA	SSB	$SSAB$	SSE
Deg freedom	1	1	1	$2^m(n-1)$
Mean square	$s_a^2 = SSA/1$	$s_b^2 = SSB/1$	$s_{ab}^2 = SSAB/1$	$s_e^2 = SSE/[2^m(n-1)]$
Computed F	$F_a = s_a^2/s_e^2$	$F_b = s_b^2/s_e^2$	$F_{ab} = s_{ab}^2/s_e^2$	
Tabulated F	$F_{[1-\alpha;1,2^m(n-1)]}$	$F_{[1-\alpha;1,2^m(n-1)]}$	$F_{[1-\alpha;1,2^m(n-1)]}$	



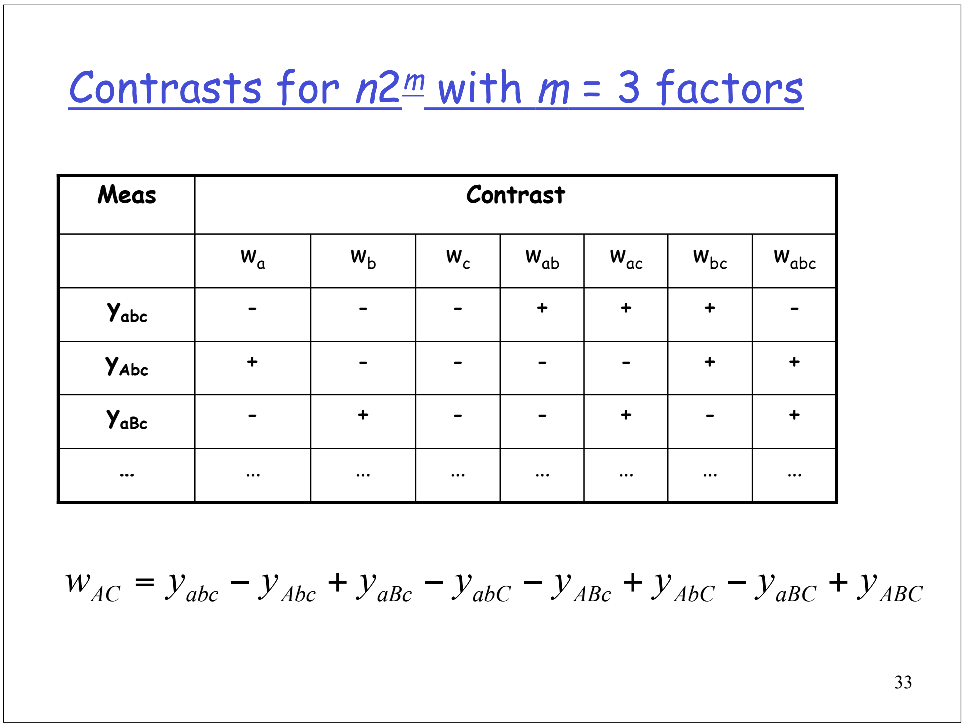
Contrasts for $n2^m$ with m = 2 factors -- revisited

Measurements	**Contrast**		
	w_a	w_b	w_{ab}
y_{AB}	+	+	+
y_{Ab}	+	-	-
y_{aB}	-	+	-
y_{ab}	-	-	+

$$w_A = y_{AB} + y_{Ab} - y_{aB} - y_{ab}$$

$$w_B = y_{AB} - y_{Ab} + y_{aB} - y_{ab}$$

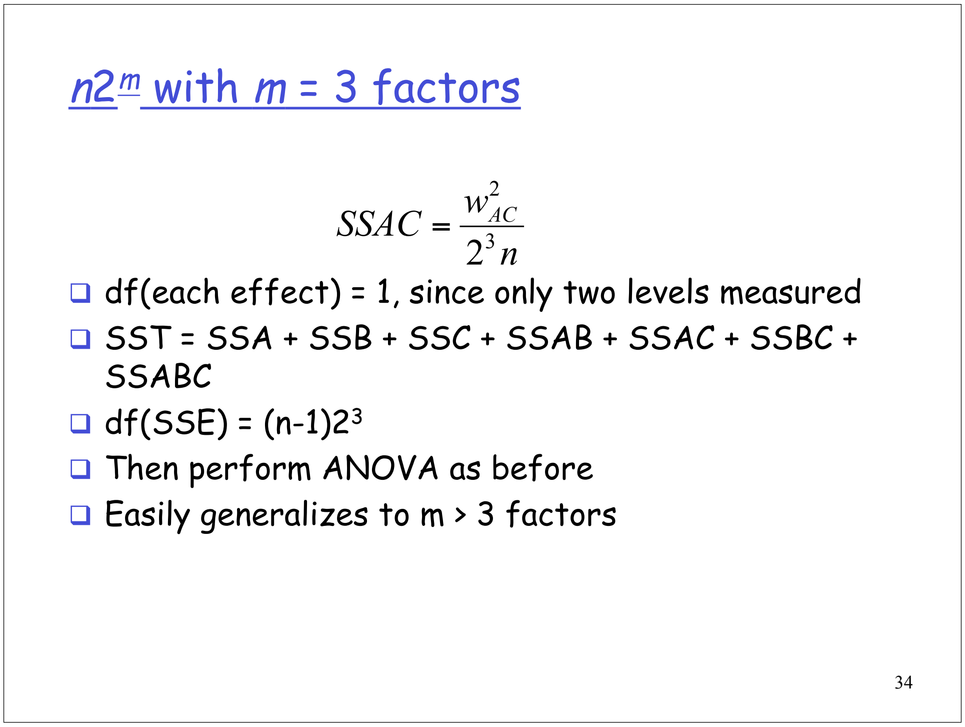
$$w_{AB} = y_{AB} - y_{Ab} - y_{aB} + y_{ab}$$



Contrasts for $n2^m$ with $m = 3$ factors

Meas	Contrast						
	w_a	w_b	w_c	w_{ab}	w_{ac}	w_{bc}	w_{abc}
y_{abc}	-	-	-	+	+	+	-
y_{Abc}	+	-	-	-	-	+	+
y_{aBc}	-	+	-	-	+	-	+
…	…	…	…	…	…	…	…

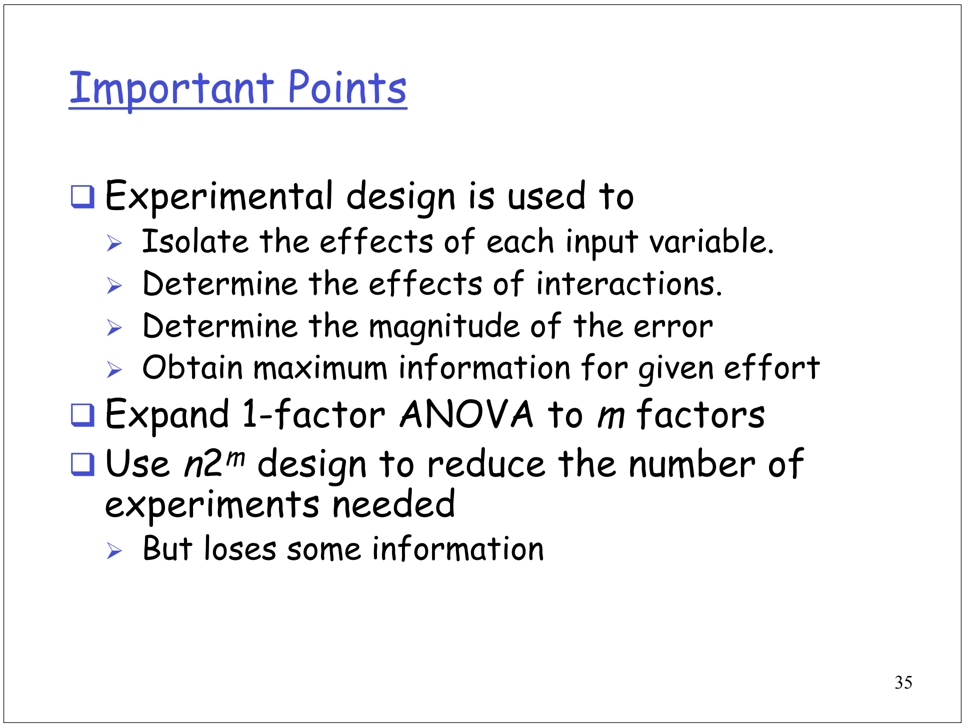
$$w_{AC} = y_{abc} - y_{Abc} + y_{aBc} - y_{abC} - y_{ABc} + y_{AbC} - y_{aBC} + y_{ABC}$$



$n2^m$ with $m = 3$ factors

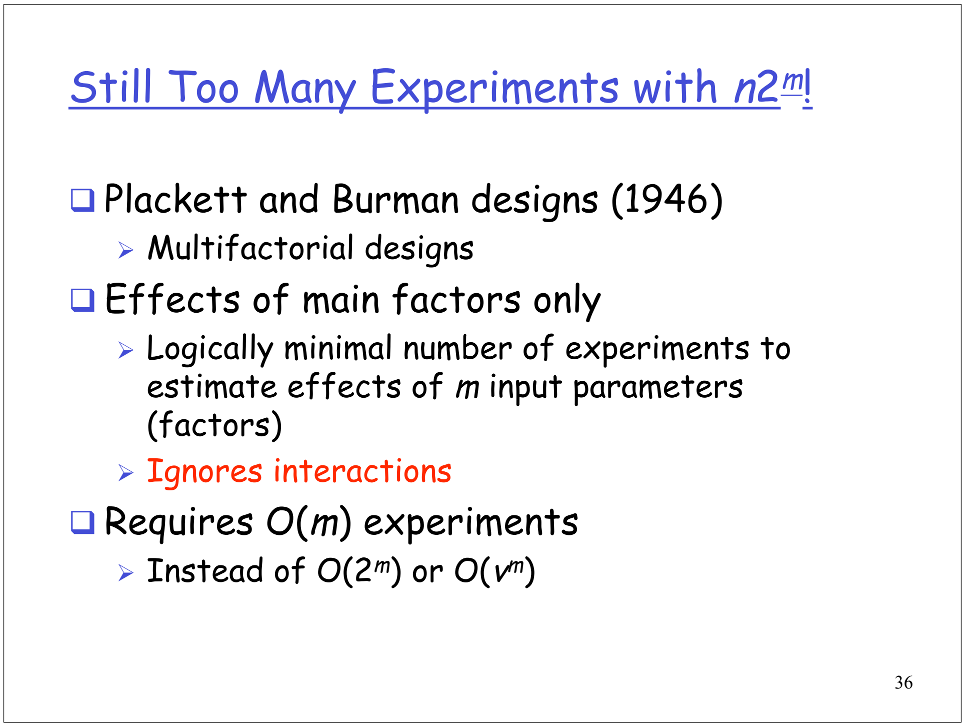
$$SSAC = \frac{w_{AC}^2}{2^3 n}$$

- ❑ df(each effect) = 1, since only two levels measured
- ❑ SST = SSA + SSB + SSC + SSAB + SSAC + SSBC + SSABC
- ❑ df(SSE) = $(n-1)2^3$
- ❑ Then perform ANOVA as before
- ❑ Easily generalizes to m > 3 factors



Important Points

- ❑ Experimental design is used to
  - Isolate the effects of each input variable.
  - Determine the effects of interactions.
  - Determine the magnitude of the error
  - Obtain maximum information for given effort
- ❑ Expand 1-factor ANOVA to *m* factors
- ❑ Use $n2^m$ design to reduce the number of experiments needed
  - But loses some information



Still Too Many Experiments with $n2^m$!

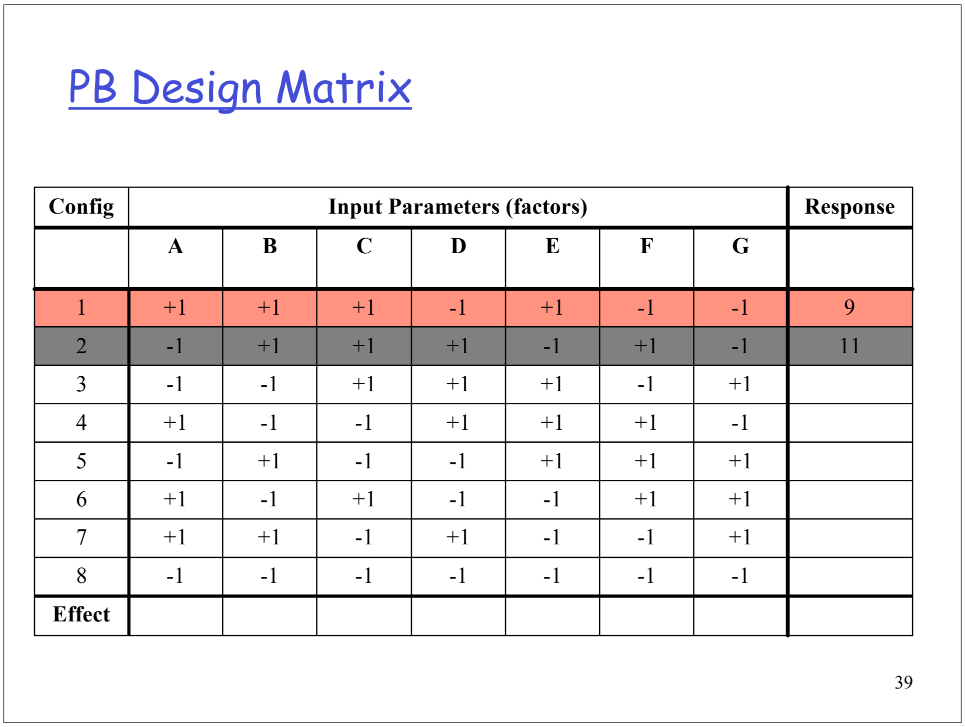
- ❑ Plackett and Burman designs (1946)
  - Multifactorial designs
- ❑ Effects of main factors only
  - Logically minimal number of experiments to estimate effects of *m* input parameters (factors)
  - Ignores interactions
- ❑ Requires $O(m)$ experiments
  - Instead of $O(2^m)$ or $O(v^m)$

Plackett and Burman Designs

- PB designs exist only in sizes that are multiples of 4
- Requires *X* experiments for *m* parameters
  - *X* = next multiple of 4 ≥ *m*
- PB design matrix
  - Rows = configurations
  - Columns = parameters' values in each config
    - High/low = +1/ -1
  - First row = from P&B paper
  - Subsequent rows = circular right shift of preceding row
  - Last row = all (-1)

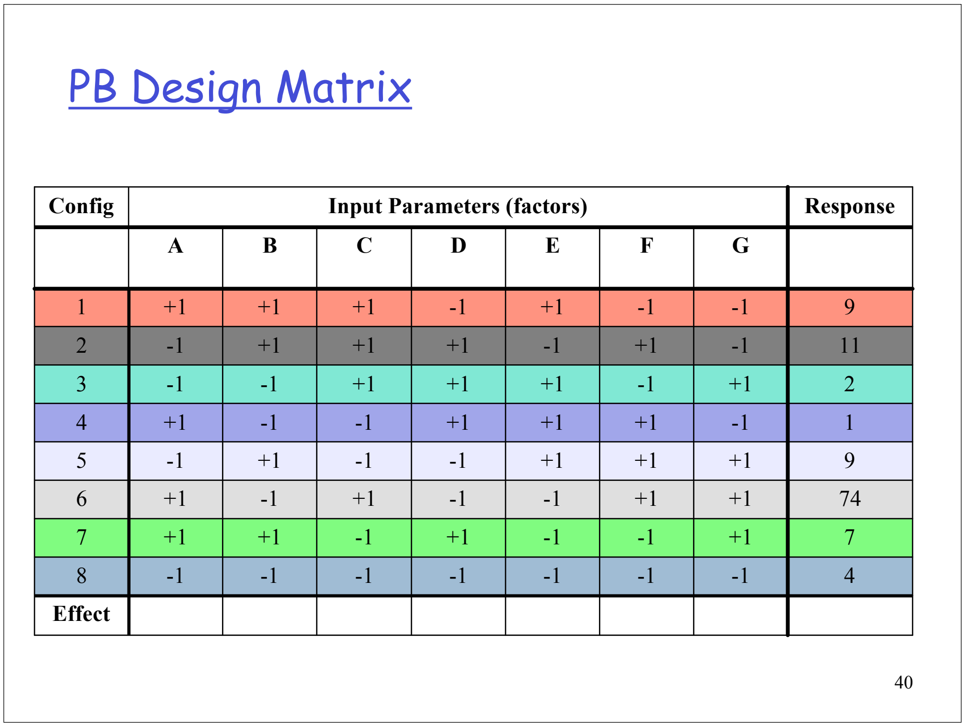
PB Design Matrix

Config	Input Parameters (factors)							Response
	A	B	C	D	E	F	G	
1	+1	+1	+1	-1	+1	-1	-1	9
2	-1	+1	+1	+1	-1	+1	-1	
3	-1	-1	+1	+1	+1	-1	+1	
4	+1	-1	-1	+1	+1	+1	-1	
5	-1	+1	-1	-1	+1	+1	+1	
6	+1	-1	+1	-1	-1	+1	+1	
7	+1	+1	-1	+1	-1	-1	+1	
8	-1	-1	-1	-1	-1	-1	-1	
Effect								



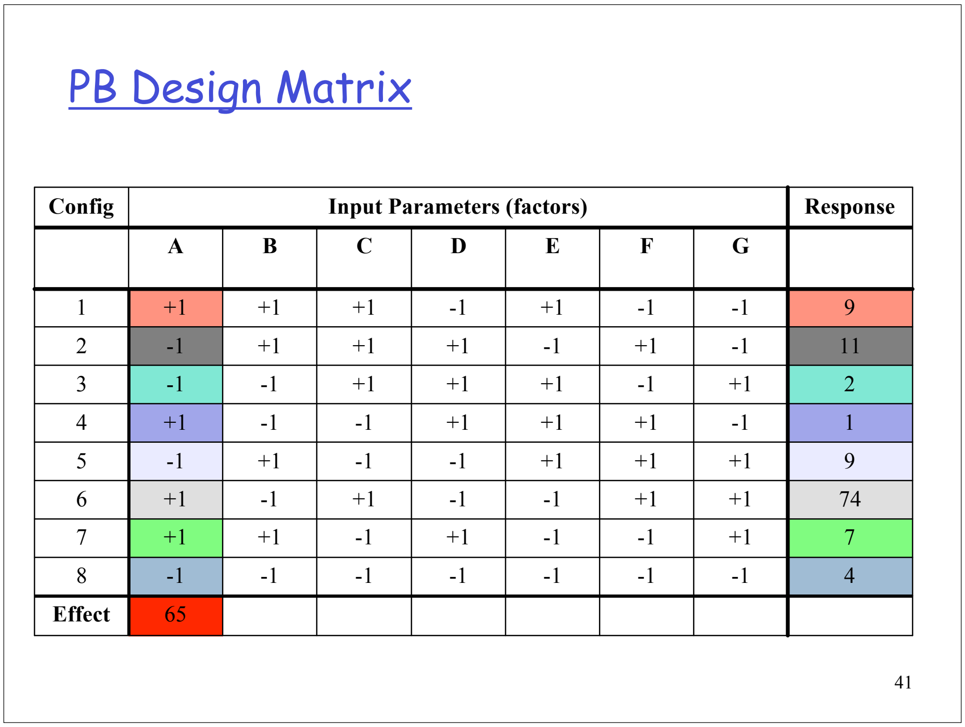
PB Design Matrix

Config	Input Parameters (factors)							Response
	A	B	C	D	E	F	G	
1	+1	+1	+1	-1	+1	-1	-1	9
2	-1	+1	+1	+1	-1	+1	-1	11
3	-1	-1	+1	+1	+1	-1	+1	
4	+1	-1	-1	+1	+1	+1	-1	
5	-1	+1	-1	-1	+1	+1	+1	
6	+1	-1	+1	-1	-1	+1	+1	
7	+1	+1	-1	+1	-1	-1	+1	
8	-1	-1	-1	-1	-1	-1	-1	
Effect								



PB Design Matrix

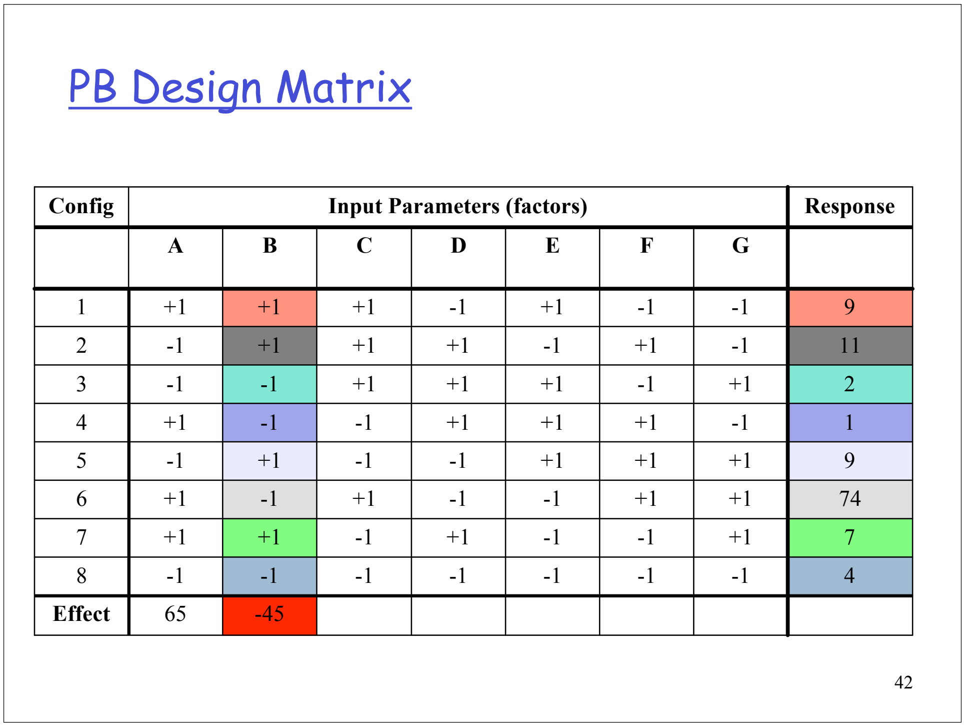
Config	Input Parameters (factors)							Response
	A	B	C	D	E	F	G	
1	+1	+1	+1	-1	+1	-1	-1	9
2	-1	+1	+1	+1	-1	+1	-1	11
3	-1	-1	+1	+1	+1	-1	+1	2
4	+1	-1	-1	+1	+1	+1	-1	1
5	-1	+1	-1	-1	+1	+1	+1	9
6	+1	-1	+1	-1	-1	+1	+1	74
7	+1	+1	-1	+1	-1	-1	+1	7
8	-1	-1	-1	-1	-1	-1	-1	4
Effect								



PB Design Matrix

Config	Input Parameters (factors)							Response
	A	B	C	D	E	F	G	
1	+1	+1	+1	-1	+1	-1	-1	9
2	-1	+1	+1	+1	-1	+1	-1	11
3	-1	-1	+1	+1	+1	-1	+1	2
4	+1	-1	-1	+1	+1	+1	-1	1
5	-1	+1	-1	-1	+1	+1	+1	9
6	+1	-1	+1	-1	-1	+1	+1	74
7	+1	+1	-1	+1	-1	-1	+1	7
8	-1	-1	-1	-1	-1	-1	-1	4
Effect	65							

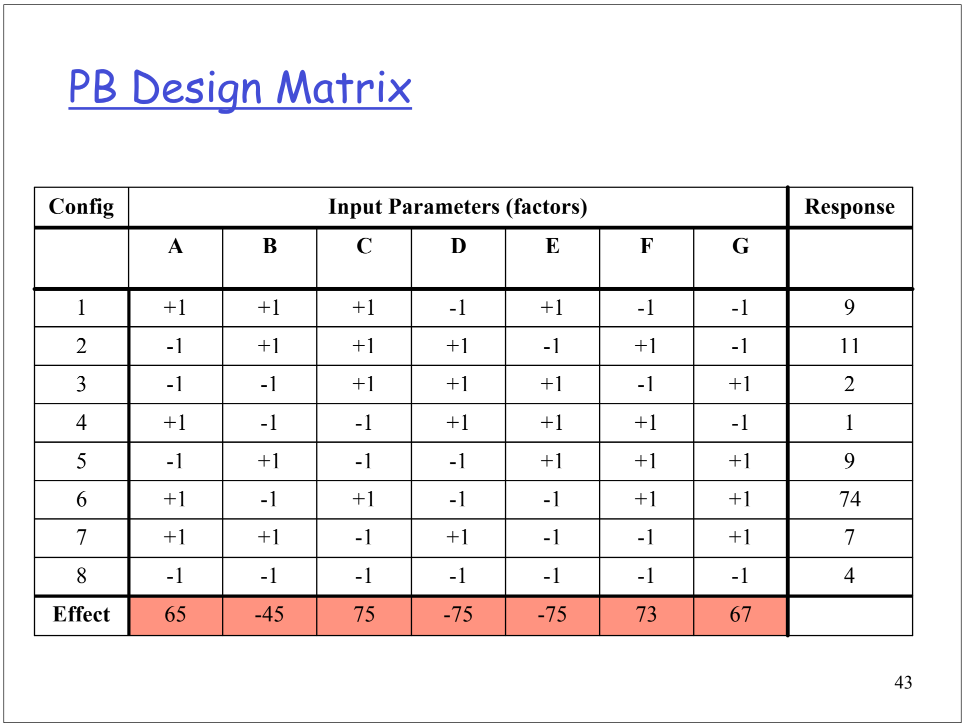
41



PB Design Matrix

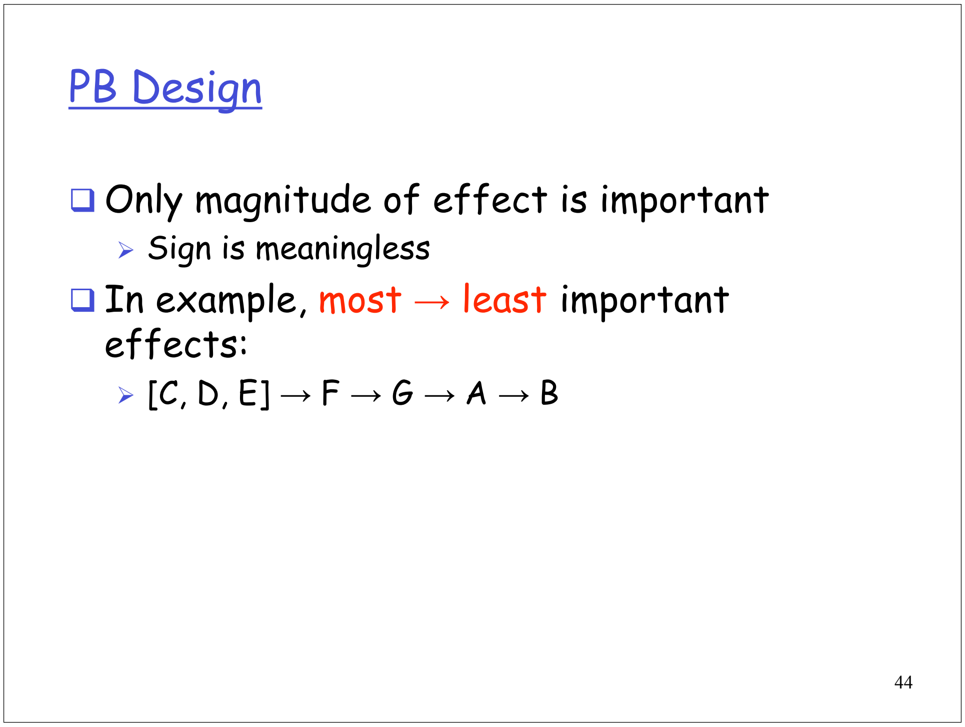
Config	Input Parameters (factors)							Response
	A	B	C	D	E	F	G	
1	+1	+1	+1	-1	+1	-1	-1	9
2	-1	+1	+1	+1	-1	+1	-1	11
3	-1	-1	+1	+1	+1	-1	+1	2
4	+1	-1	-1	+1	+1	+1	-1	1
5	-1	+1	-1	-1	+1	+1	+1	9
6	+1	-1	+1	-1	-1	+1	+1	74
7	+1	+1	-1	+1	-1	-1	+1	7
8	-1	-1	-1	-1	-1	-1	-1	4
Effect	65	-45						

42



PB Design Matrix

Config	Input Parameters (factors)							Response
	A	B	C	D	E	F	G	
1	+1	+1	+1	-1	+1	-1	-1	9
2	-1	+1	+1	+1	-1	+1	-1	11
3	-1	-1	+1	+1	+1	-1	+1	2
4	+1	-1	-1	+1	+1	+1	-1	1
5	-1	+1	-1	-1	+1	+1	+1	9
6	+1	-1	+1	-1	-1	+1	+1	74
7	+1	+1	-1	+1	-1	-1	+1	7
8	-1	-1	-1	-1	-1	-1	-1	4
Effect	65	-45	75	-75	-75	73	67	



PB Design

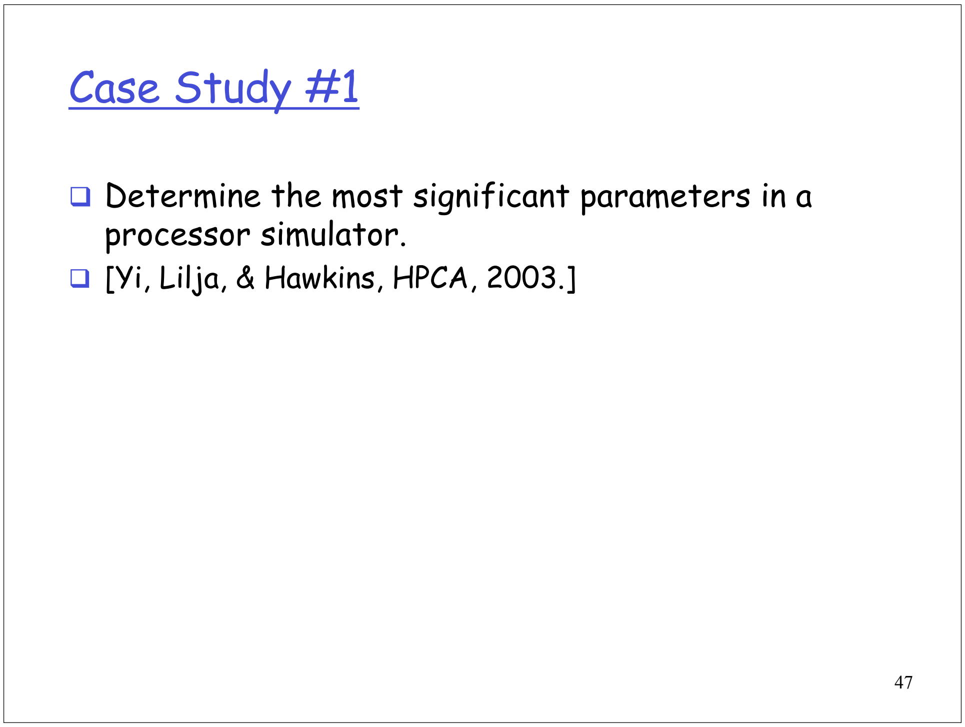
- ❑ Only magnitude of effect is important
  - ➢ Sign is meaningless
- ❑ In example, most → least important effects:
  - ➢ [C, D, E] → F → G → A → B

PB Design Matrix with Foldover

- ❑ Add *X* additional rows to matrix
  - ➢ Signs of additional rows are opposite original rows
- ❑ Provides some additional information about selected interactions

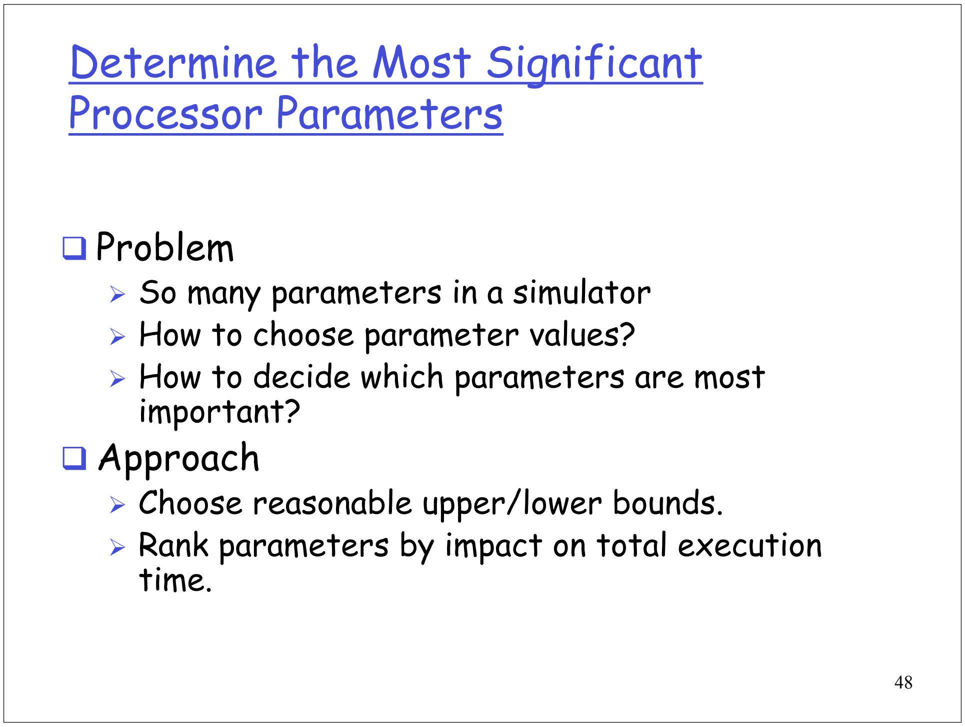
PB Design Matrix with Foldover

A	B	C	D	E	F	G	Exec. Time
+1	+1	+1	-1	+1	-1	-1	9
-1	+1	+1	+1	-1	+1	-1	11
-1	-1	+1	+1	+1	-1	+1	2
+1	-1	-1	+1	+1	+1	-1	1
-1	+1	-1	-1	+1	+1	+1	9
+1	-1	+1	-1	-1	+1	+1	74
+1	+1	-1	+1	-1	-1	+1	7
-1	-1	-1	-1	-1	-1	-1	4
-1	-1	-1	+1	-1	+1	+1	17
+1	-1	-1	-1	+1	-1	+1	76
+1	+1	-1	-1	-1	+1	-1	6
-1	+1	+1	-1	-1	-1	+1	31
+1	-1	+1	+1	-1	-1	-1	19
-1	+1	-1	+1	+1	-1	-1	33
-1	-1	+1	-1	+1	+1	-1	6
+1	+1	+1	+1	+1	+1	+1	112
191	19	111	-13	79	55	239	



Case Study #1

- Determine the most significant parameters in a processor simulator.
- [Yi, Lilja, & Hawkins, HPCA, 2003.]



Determine the Most Significant Processor Parameters

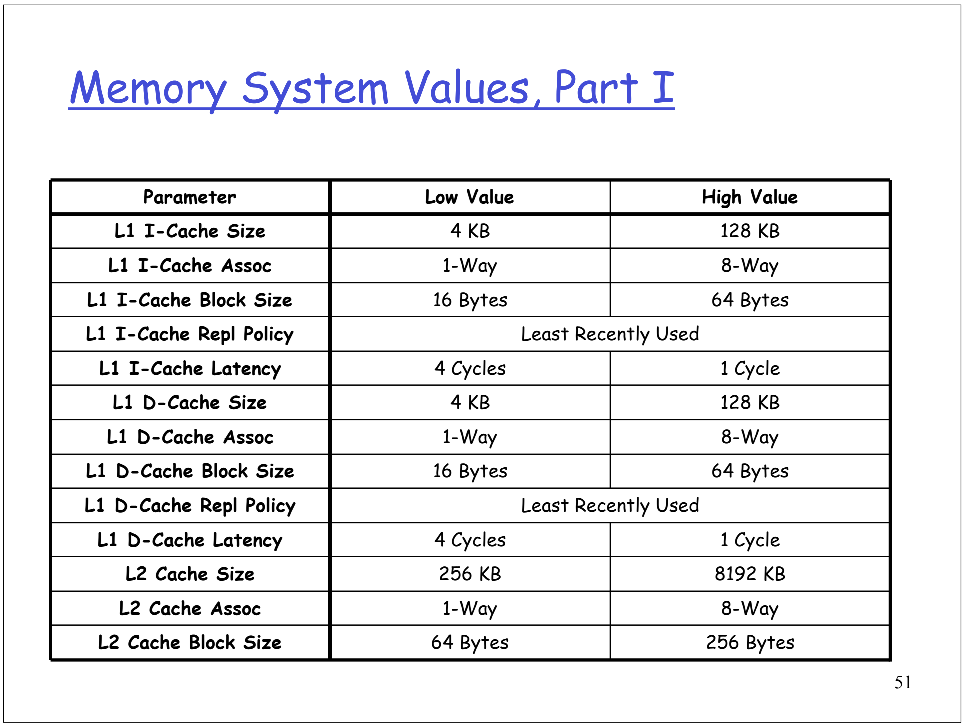
- Problem
  - So many parameters in a simulator
  - How to choose parameter values?
  - How to decide which parameters are most important?
- Approach
  - Choose reasonable upper/lower bounds.
  - Rank parameters by impact on total execution time.

Simulation Environment

- SimpleScalar simulator
  - sim-outorder 3.0
- Selected SPEC 2000 Benchmarks
  - *gzip, vpr, gcc, mesa, art, mcf, equake, parser, vortex, bzip2, twolf*
- MinneSPEC Reduced Input Sets
- Compiled with gcc (PISA) at O3

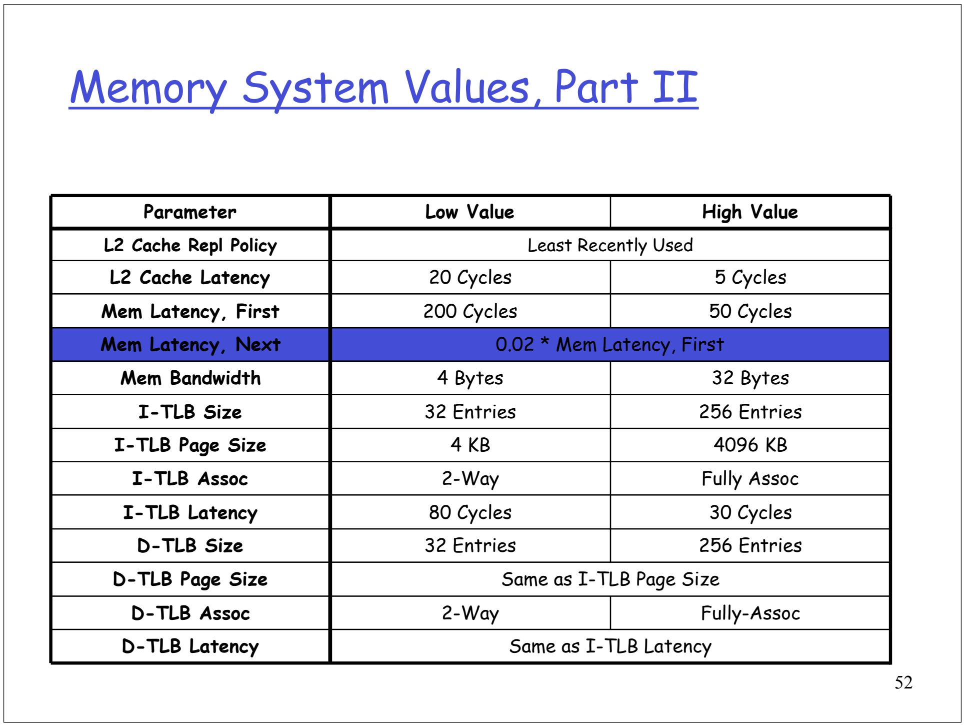
Functional Unit Values

Parameter	Low Value	High Value
Int ALUs	1	4
Int ALU Latency	2 Cycles	1 Cycle
Int ALU Throughput	1	
FP ALUs	1	4
FP ALU Latency	5 Cycles	1 Cycle
FP ALU Throughputs	1	
Int Mult/Div Units	1	4
Int Mult Latency	15 Cycles	2 Cycles
Int Div Latency	80 Cycles	10 Cycles
Int Mult Throughput	1	
Int Div Throughput	Equal to Int Div Latency	
FP Mult/Div Units	1	4
FP Mult Latency	5 Cycles	2 Cycles
FP Div Latency	35 Cycles	10 Cycles
FP Sqrt Latency	35 Cycles	15 Cycles
FP Mult Throughput	Equal to FP Mult Latency	
FP Div Throughput	Equal to FP Div Latency	
FP Sqrt Throughput	Equal to FP Sqrt Latency	



Memory System Values, Part I

Parameter	Low Value	High Value
L1 I-Cache Size	4 KB	128 KB
L1 I-Cache Assoc	1-Way	8-Way
L1 I-Cache Block Size	16 Bytes	64 Bytes
L1 I-Cache Repl Policy	Least Recently Used	
L1 I-Cache Latency	4 Cycles	1 Cycle
L1 D-Cache Size	4 KB	128 KB
L1 D-Cache Assoc	1-Way	8-Way
L1 D-Cache Block Size	16 Bytes	64 Bytes
L1 D-Cache Repl Policy	Least Recently Used	
L1 D-Cache Latency	4 Cycles	1 Cycle
L2 Cache Size	256 KB	8192 KB
L2 Cache Assoc	1-Way	8-Way
L2 Cache Block Size	64 Bytes	256 Bytes



Memory System Values, Part II

Parameter	Low Value	High Value
L2 Cache Repl Policy	Least Recently Used	
L2 Cache Latency	20 Cycles	5 Cycles
Mem Latency, First	200 Cycles	50 Cycles
Mem Latency, Next	0.02 * Mem Latency, First	
Mem Bandwidth	4 Bytes	32 Bytes
I-TLB Size	32 Entries	256 Entries
I-TLB Page Size	4 KB	4096 KB
I-TLB Assoc	2-Way	Fully Assoc
I-TLB Latency	80 Cycles	30 Cycles
D-TLB Size	32 Entries	256 Entries
D-TLB Page Size	Same as I-TLB Page Size	
D-TLB Assoc	2-Way	Fully-Assoc
D-TLB Latency	Same as I-TLB Latency	

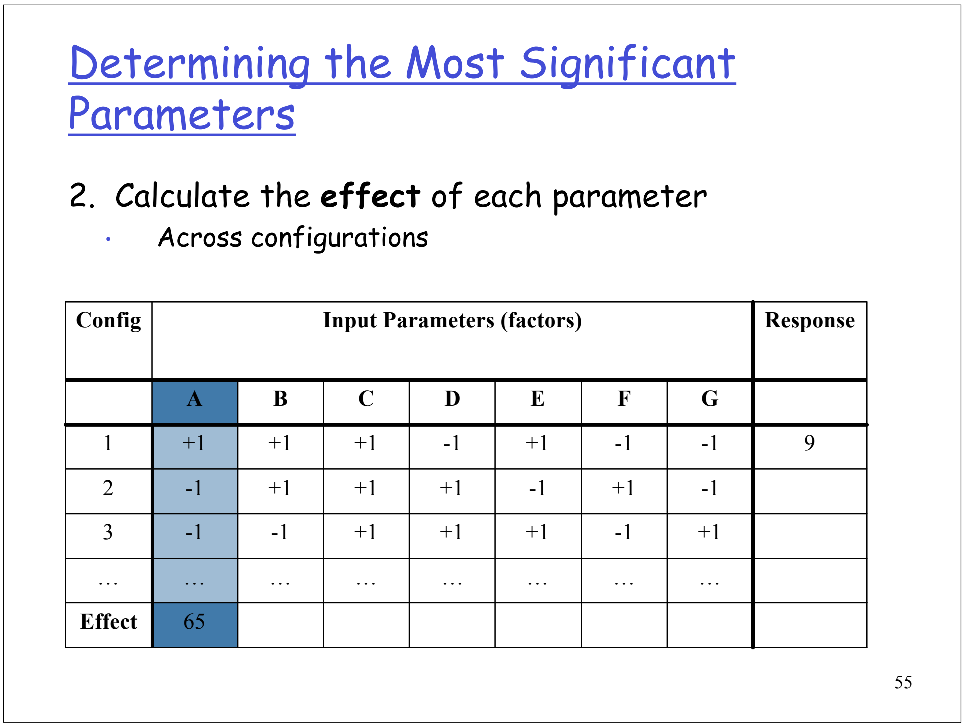
Processor Core Values

Parameter	Low Value	High Value
Fetch Queue Entries	4	32
Branch Predictor	2-Level	Perfect
Branch MPred Penalty	10 Cycles	2 Cycles
RAS Entries	4	64
BTB Entries	16	512
BTB Assoc	2-Way	Fully-Assoc
Spec Branch Update	In Commit	In Decode
Decode/Issue Width	4-Way	
ROB Entries	8	64
LSQ Entries	0.25 * ROB	1.0 * ROB
Memory Ports	1	4

Determining the Most Significant Parameters

1. Run simulations to find **response**
   - With input parameters at high/low, on/off values

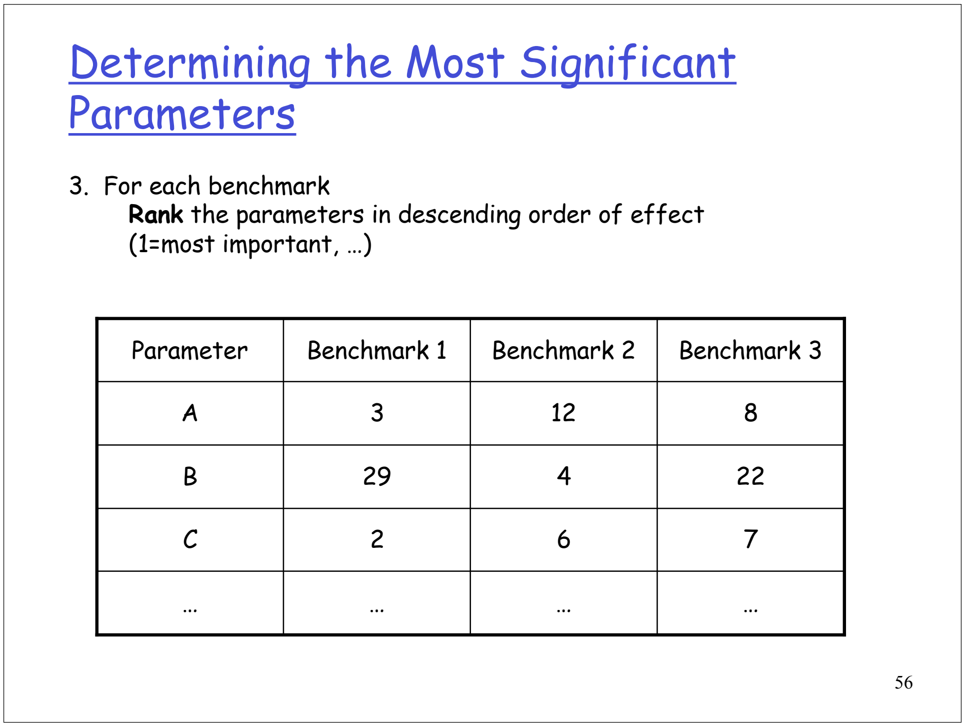
Config	Input Parameters (factors)							Response
	A	B	C	D	E	F	G	
1	+1	+1	+1	-1	+1	-1	-1	9
2	-1	+1	+1	+1	-1	+1	-1	
3	-1	-1	+1	+1	+1	-1	+1	
…	…	…	…	…	…	…	…	
Effect								



Determining the Most Significant Parameters

2. Calculate the **effect** of each parameter
   - Across configurations

Config	Input Parameters (factors)							Response
	A	B	C	D	E	F	G	
1	+1	+1	+1	-1	+1	-1	-1	9
2	-1	+1	+1	+1	-1	+1	-1	
3	-1	-1	+1	+1	+1	-1	+1	
…	…	…	…	…	…	…	…	
Effect	65							



Determining the Most Significant Parameters

3. For each benchmark
   Rank the parameters in descending order of effect
   (1=most important, …)

Parameter	Benchmark 1	Benchmark 2	Benchmark 3
A	3	12	8
B	29	4	22
C	2	6	7
…	…	…	…

Determining the Most Significant Parameters

4. For each parameter

   Average the ranks

Parameter	Benchmark 1	Benchmark 2	Benchmark 3	Average
A	3	12	8	7.67
B	29	4	22	18.3
C	2	6	7	5
…	…	…	…	…

Most Significant Parameters

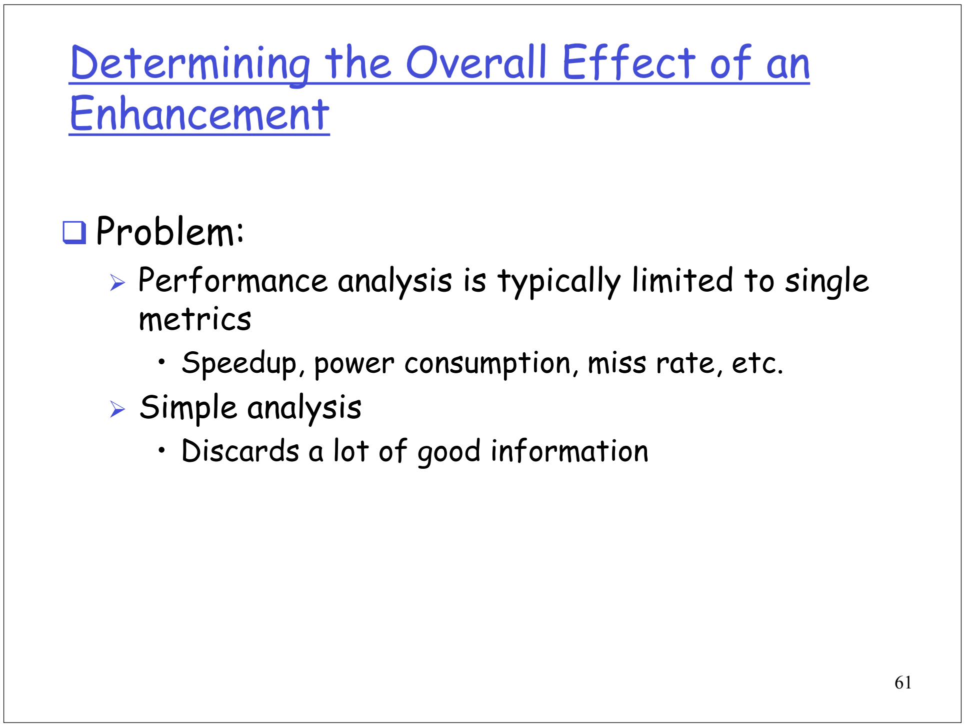
Number	Parameter	gcc	gzip	art	Average
1	ROB Entries	4	1	2	2.77
2	L2 Cache Latency	2	4	4	4.00
3	Branch Predictor Accuracy	5	2	27	7.69
4	Number of Integer ALUs	8	3	29	9.08
5	L1 D-Cache Latency	7	7	8	10.00
6	L1 I-Cache Size	1	6	12	10.23
7	L2 Cache Size	6	9	1	10.62
8	L1 I-Cache Block Size	3	16	10	11.77
9	Memory Latency, First	9	36	3	12.31
10	LSQ Entries	10	12	39	12.62
11	Speculative Branch Update	28	8	16	18.23

General Procedure

- Determine upper/lower *bounds* for parameters
- Simulate configurations to find *response*
- Compute *effects* of each parameter for each configuration
- *Rank* the parameters for each benchmark based on effects
- *Average* the ranks across benchmarks
- Focus on *top-ranked* parameters for subsequent analysis

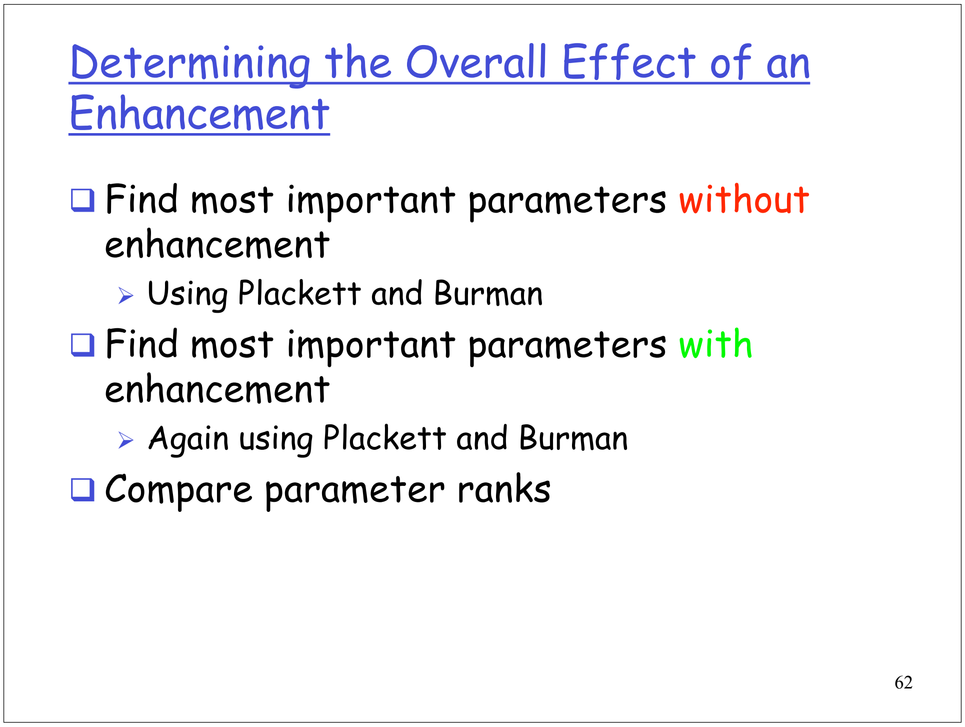
Case Study #2

- Determine the "big picture" impact of a system enhancement.



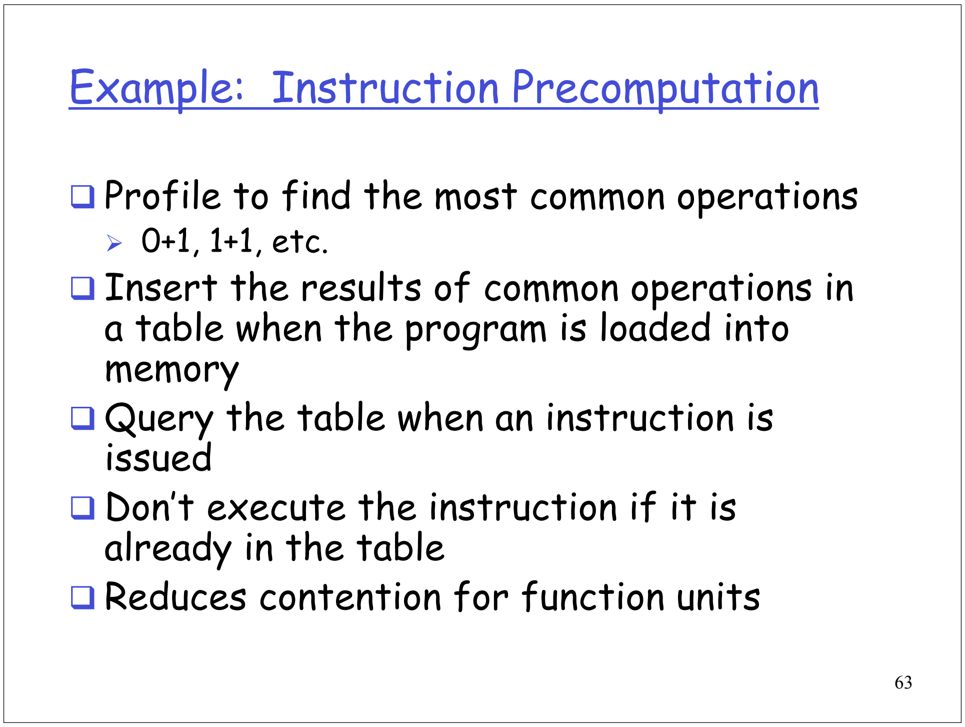
Determining the Overall Effect of an Enhancement

- ❑ Problem:
  - Performance analysis is typically limited to single metrics
    - Speedup, power consumption, miss rate, etc.
  - Simple analysis
    - Discards a lot of good information



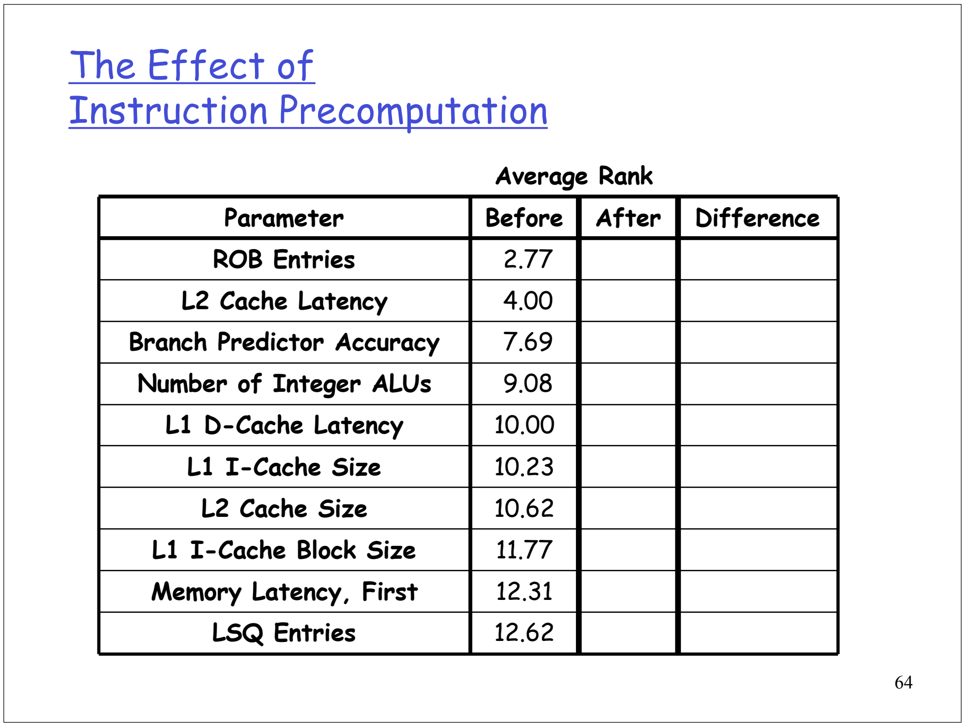
Determining the Overall Effect of an Enhancement

- ❑ Find most important parameters without enhancement
  - Using Plackett and Burman
- ❑ Find most important parameters with enhancement
  - Again using Plackett and Burman
- ❑ Compare parameter ranks



Example: Instruction Precomputation

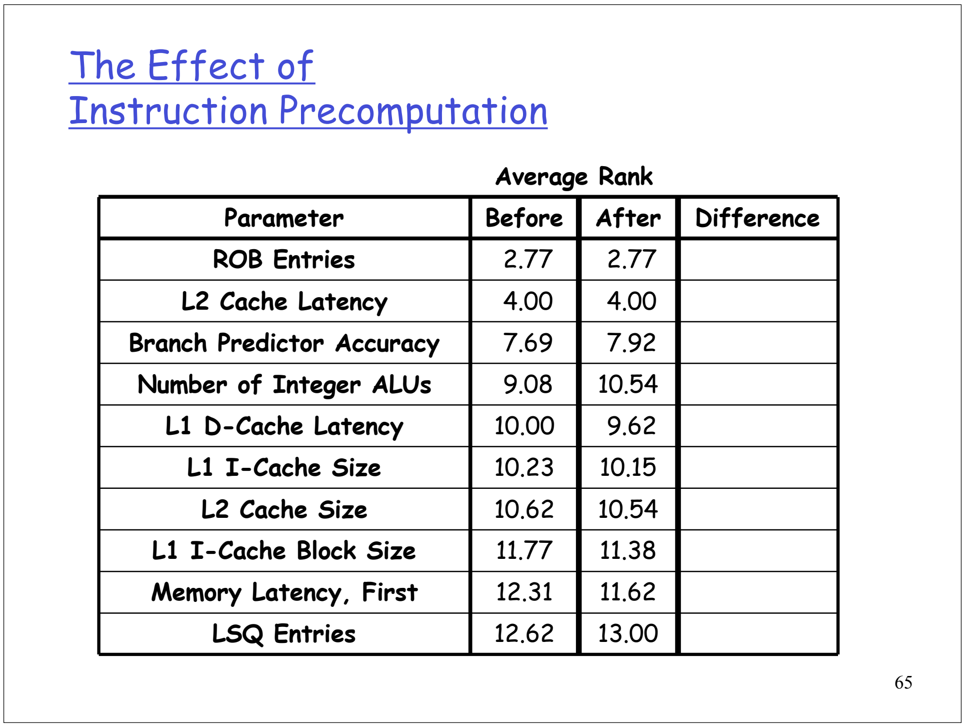
- Profile to find the most common operations
  - 0+1, 1+1, etc.
- Insert the results of common operations in a table when the program is loaded into memory
- Query the table when an instruction is issued
- Don't execute the instruction if it is already in the table
- Reduces contention for function units



The Effect of Instruction Precomputation

Average Rank

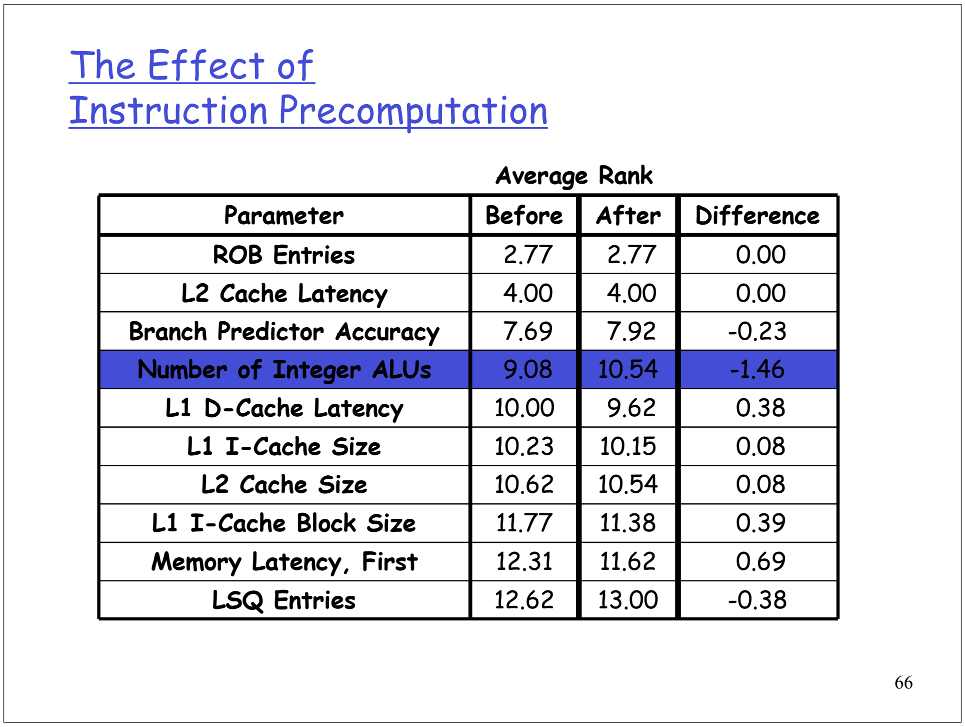
Parameter	Before	After	Difference
ROB Entries	2.77		
L2 Cache Latency	4.00		
Branch Predictor Accuracy	7.69		
Number of Integer ALUs	9.08		
L1 D-Cache Latency	10.00		
L1 I-Cache Size	10.23		
L2 Cache Size	10.62		
L1 I-Cache Block Size	11.77		
Memory Latency, First	12.31		
LSQ Entries	12.62		



The Effect of Instruction Precomputation

Average Rank

Parameter	Before	After	Difference
ROB Entries	2.77	2.77	
L2 Cache Latency	4.00	4.00	
Branch Predictor Accuracy	7.69	7.92	
Number of Integer ALUs	9.08	10.54	
L1 D-Cache Latency	10.00	9.62	
L1 I-Cache Size	10.23	10.15	
L2 Cache Size	10.62	10.54	
L1 I-Cache Block Size	11.77	11.38	
Memory Latency, First	12.31	11.62	
LSQ Entries	12.62	13.00	



The Effect of Instruction Precomputation

Average Rank

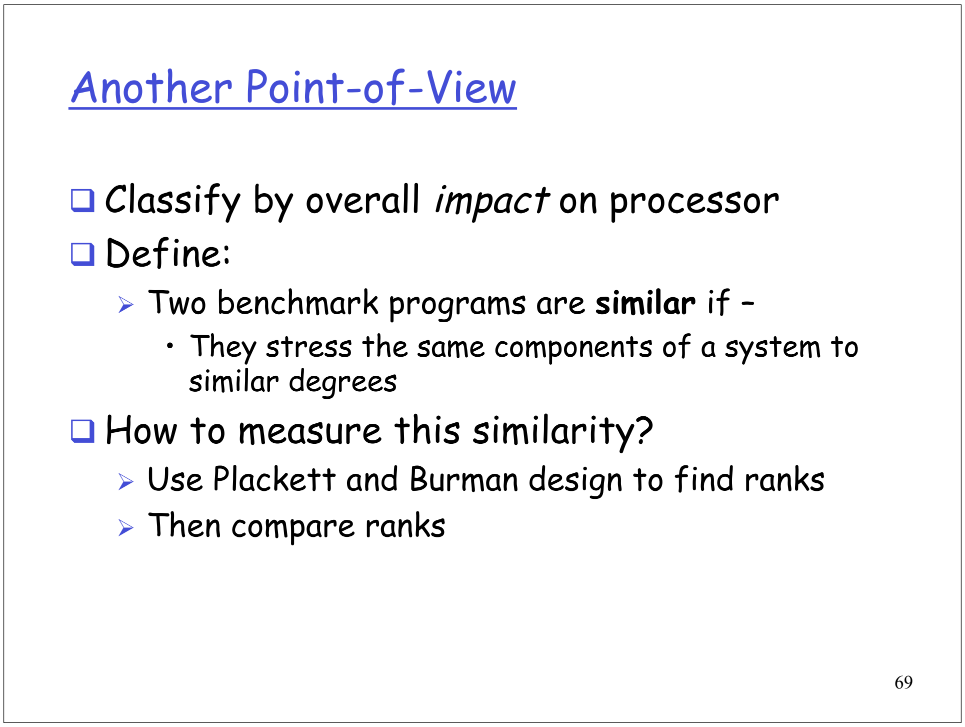
Parameter	Before	After	Difference
ROB Entries	2.77	2.77	0.00
L2 Cache Latency	4.00	4.00	0.00
Branch Predictor Accuracy	7.69	7.92	-0.23
Number of Integer ALUs	9.08	10.54	-1.46
L1 D-Cache Latency	10.00	9.62	0.38
L1 I-Cache Size	10.23	10.15	0.08
L2 Cache Size	10.62	10.54	0.08
L1 I-Cache Block Size	11.77	11.38	0.39
Memory Latency, First	12.31	11.62	0.69
LSQ Entries	12.62	13.00	-0.38

Case Study #3

- ❑ Benchmark program classification.

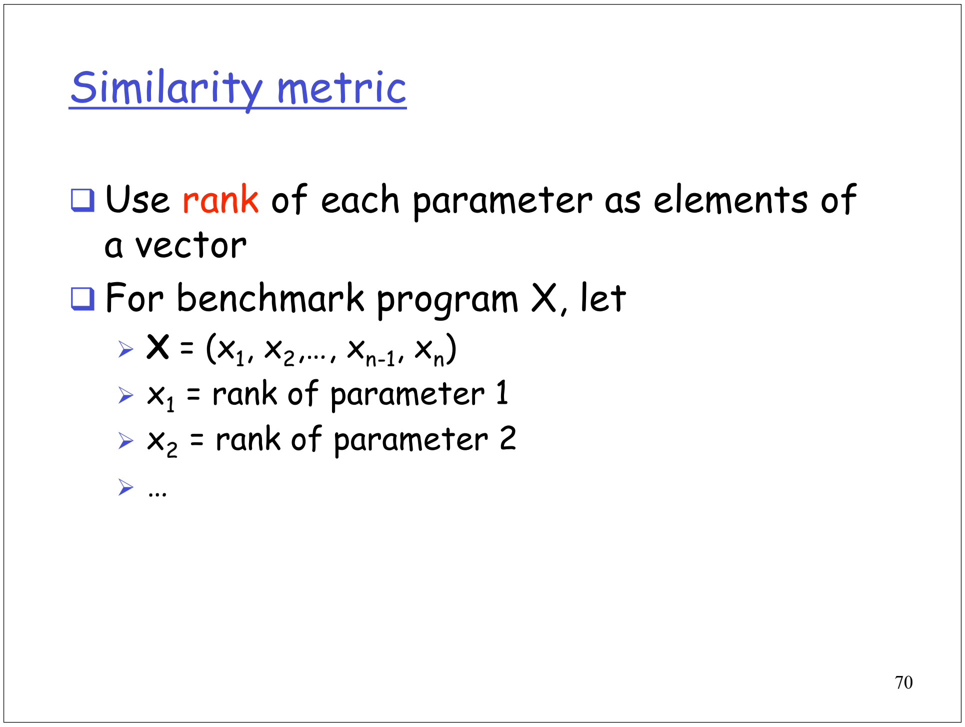
Benchmark Classification

- ❑ By application type
  - Scientific and engineering applications
  - Transaction processing applications
  - Multimedia applications
- ❑ By use of processor function units
  - Floating-point code
  - Integer code
  - Memory intensive code
- ❑ Etc., etc.



Another Point-of-View

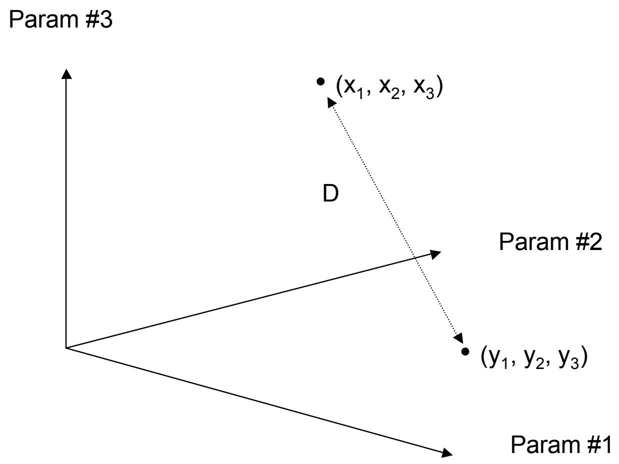
- ❑ Classify by overall *impact* on processor
- ❑ Define:
  - ➢ Two benchmark programs are **similar** if –
    - They stress the same components of a system to similar degrees
- ❑ How to measure this similarity?
  - ➢ Use Plackett and Burman design to find ranks
  - ➢ Then compare ranks



Similarity metric

- ❑ Use rank of each parameter as elements of a vector
- ❑ For benchmark program X, let
  - ➢ $\mathbf{X} = (x_1, x_2, \ldots, x_{n-1}, x_n)$
  - ➢ x_1 = rank of parameter 1
  - ➢ x_2 = rank of parameter 2
  - ➢ …

Vector Defines a Point in *n*-space



Similarity Metric

- ❑ Euclidean Distance Between Points

$$D = [(x_1 - y_1)^2 + (x_2 - y_2)^2 + \ldots + (x_{n-1} - y_{n-1})^2 + (x_n - y_n)^2]^{1/2}$$

Most Significant Parameters

Number	Parameter	gcc	gzip	art
1	ROB Entries	4	1	2
2	L2 Cache Latency	2	4	4
3	Branch Predictor Accuracy	5	2	27
4	Number of Integer ALUs	8	3	29
5	L1 D-Cache Latency	7	7	8
6	L1 I-Cache Size	1	6	12
7	L2 Cache Size	6	9	1
8	L1 I-Cache Block Size	3	16	10
9	Memory Latency, First	9	36	3
10	LSQ Entries	10	12	39
11	Speculative Branch Update	28	8	16

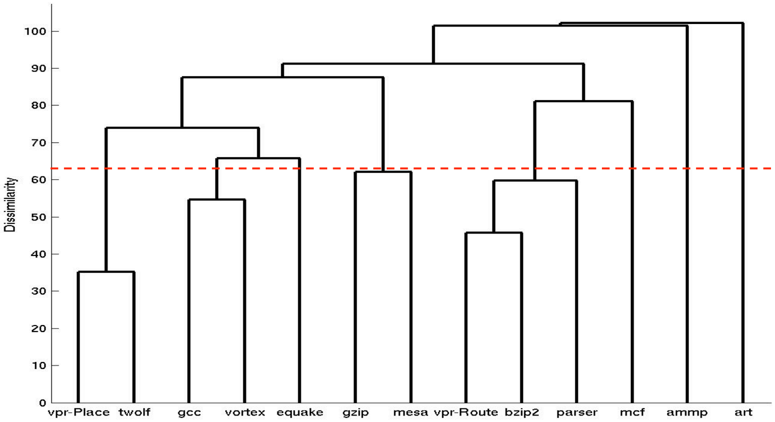
Distance Computation

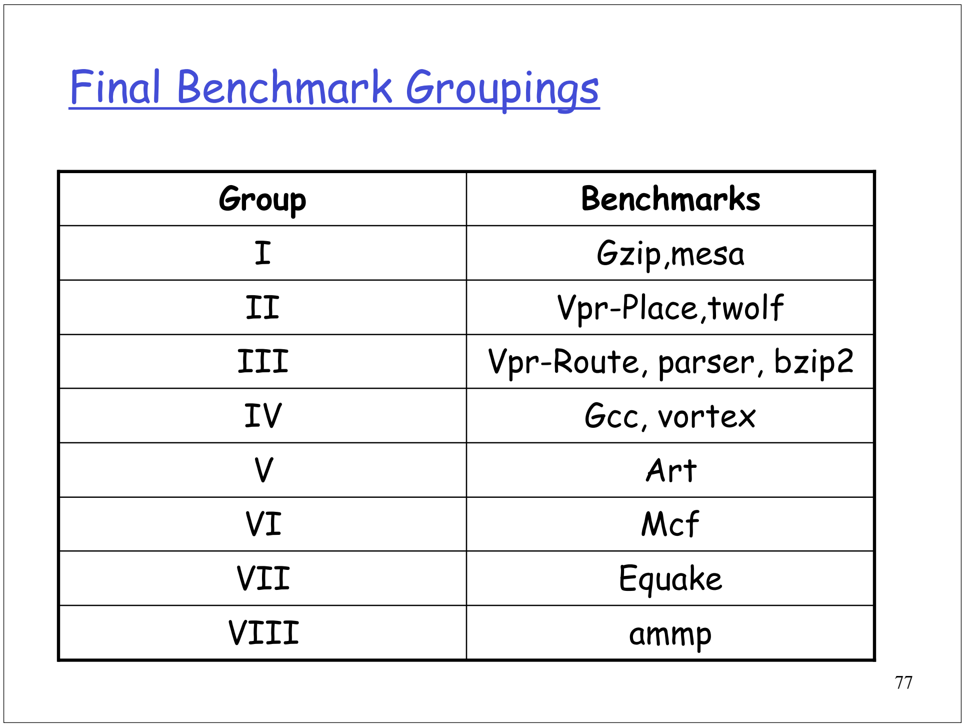
- Rank vectors
  - Gcc = (4, 2, 5, 8, …)
  - Gzip = (1, 4, 2, 3, …)
  - Art = (2, 4, 27, 29, …)
- Euclidean distances
  - D(gcc - gzip) = $[(4-1)^2 + (2-4)^2 + (5-2)^2 + \dots]^{1/2}$
  - D(gcc - art) = $[(4-2)^2 + (2-4)^2 + (5-27)^2 + \dots]^{1/2}$
  - D(gzip - art) = $[(1-2)^2 + (4-4)^2 + (2-27)^2 + \dots]^{1/2}$

Euclidean Distances for Selected Benchmarks

	gcc	gzip	art	mcf
gcc	0	81.9	92.6	94.5
gzip		0	113.5	109.6
art			0	98.6
mcf				0

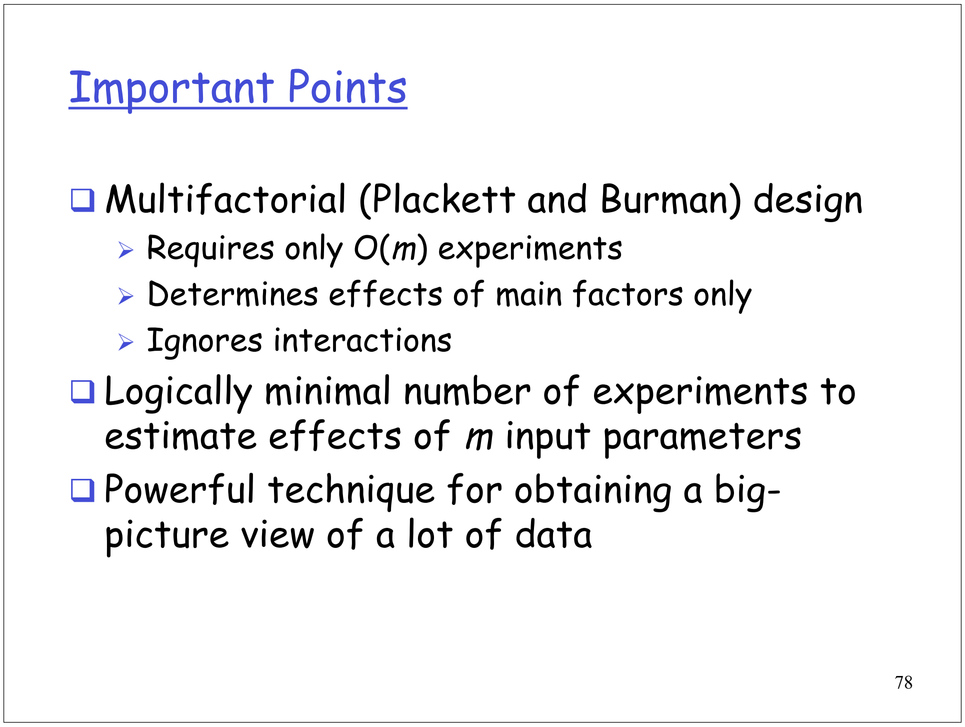
Dendogram of Distances Showing (Dis-)Similarity





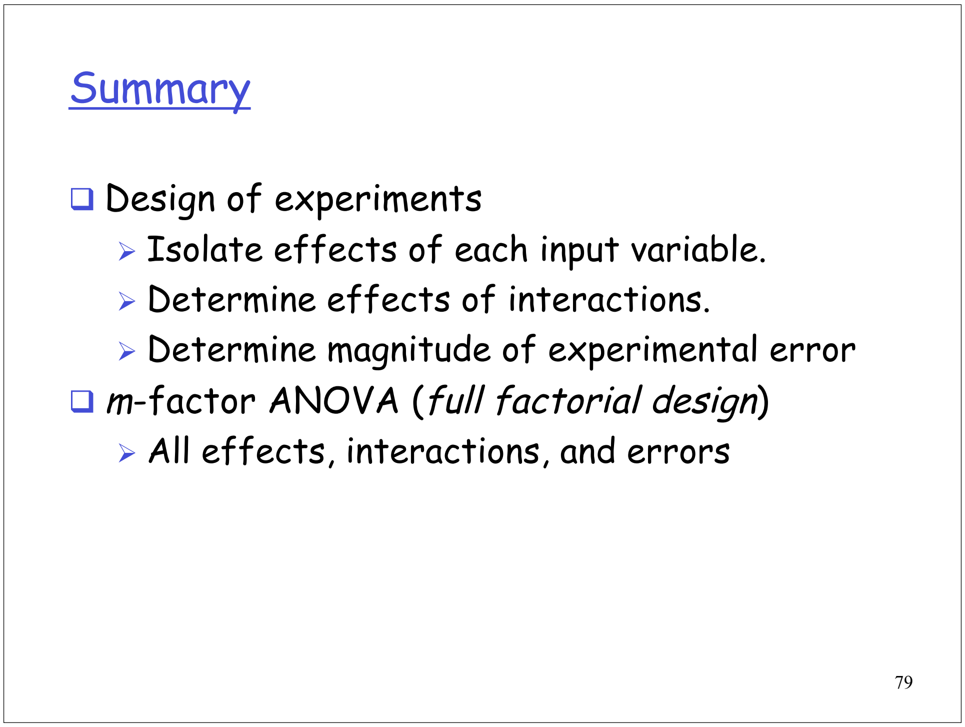
Final Benchmark Groupings

Group	Benchmarks
I	Gzip,mesa
II	Vpr-Place,twolf
III	Vpr-Route, parser, bzip2
IV	Gcc, vortex
V	Art
VI	Mcf
VII	Equake
VIII	ammp



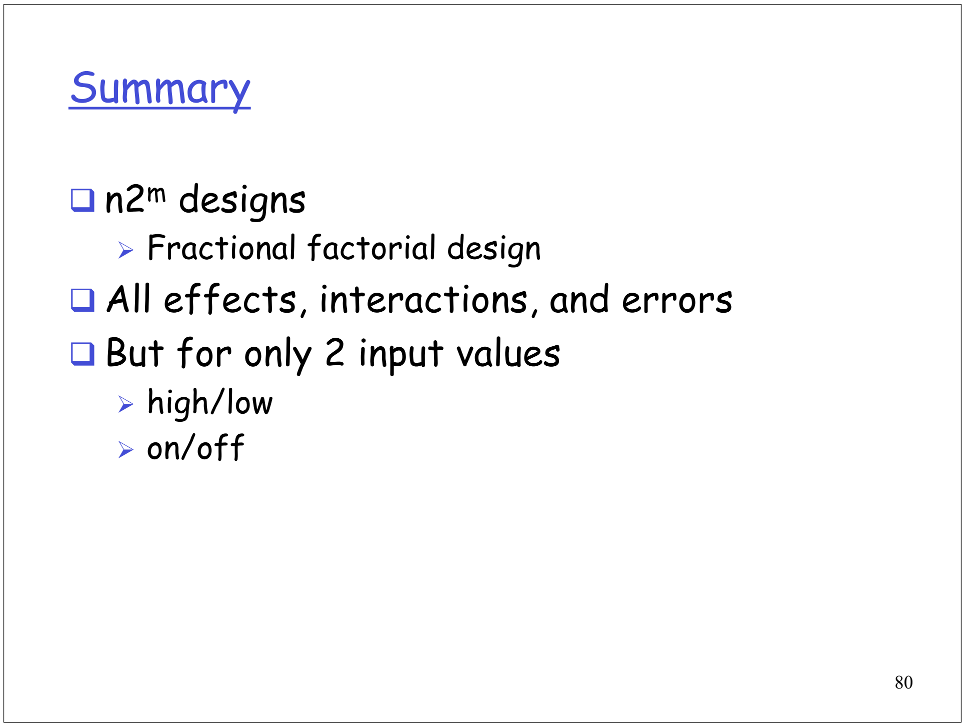
Important Points

- ❑ Multifactorial (Plackett and Burman) design
  - ➢ Requires only O(*m*) experiments
  - ➢ Determines effects of main factors only
  - ➢ Ignores interactions
- ❑ Logically minimal number of experiments to estimate effects of *m* input parameters
- ❑ Powerful technique for obtaining a big-picture view of a lot of data



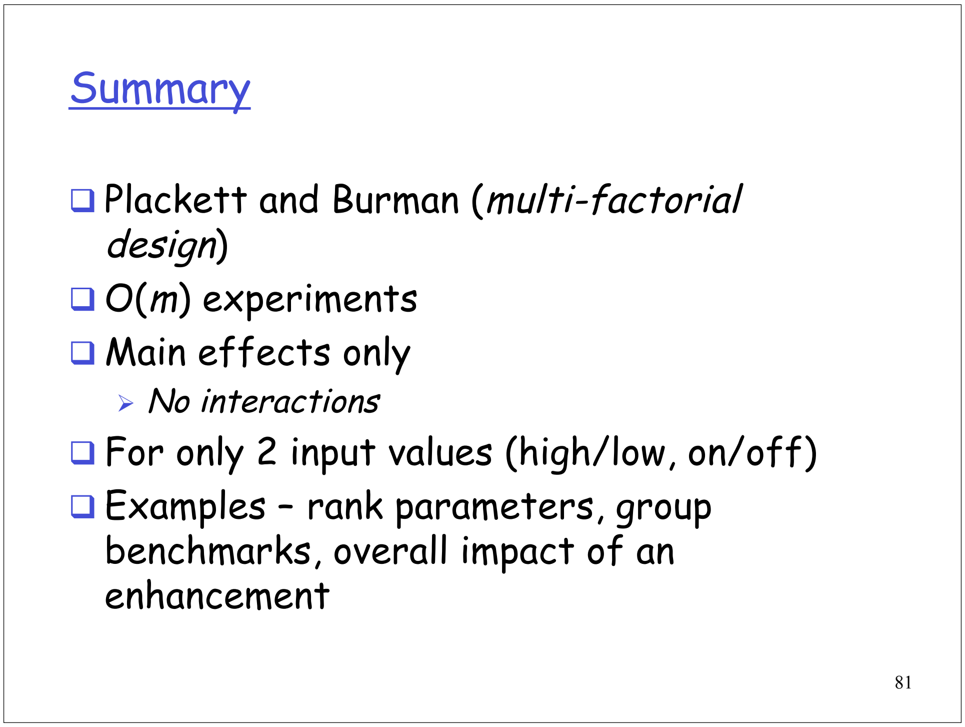
Summary

- ❑ Design of experiments
  - ➢ Isolate effects of each input variable.
  - ➢ Determine effects of interactions.
  - ➢ Determine magnitude of experimental error
- ❑ *m*-factor ANOVA (*full factorial design*)
  - ➢ All effects, interactions, and errors



Summary

- ❑ n2^m designs
  - ➢ Fractional factorial design
- ❑ All effects, interactions, and errors
- ❑ But for only 2 input values
  - ➢ high/low
  - ➢ on/off



Summary

- Plackett and Burman (*multi-factorial design*)
- O(*m*) experiments
- Main effects only
  - *No interactions*
- For only 2 input values (high/low, on/off)
- Examples – rank parameters, group benchmarks, overall impact of an enhancement