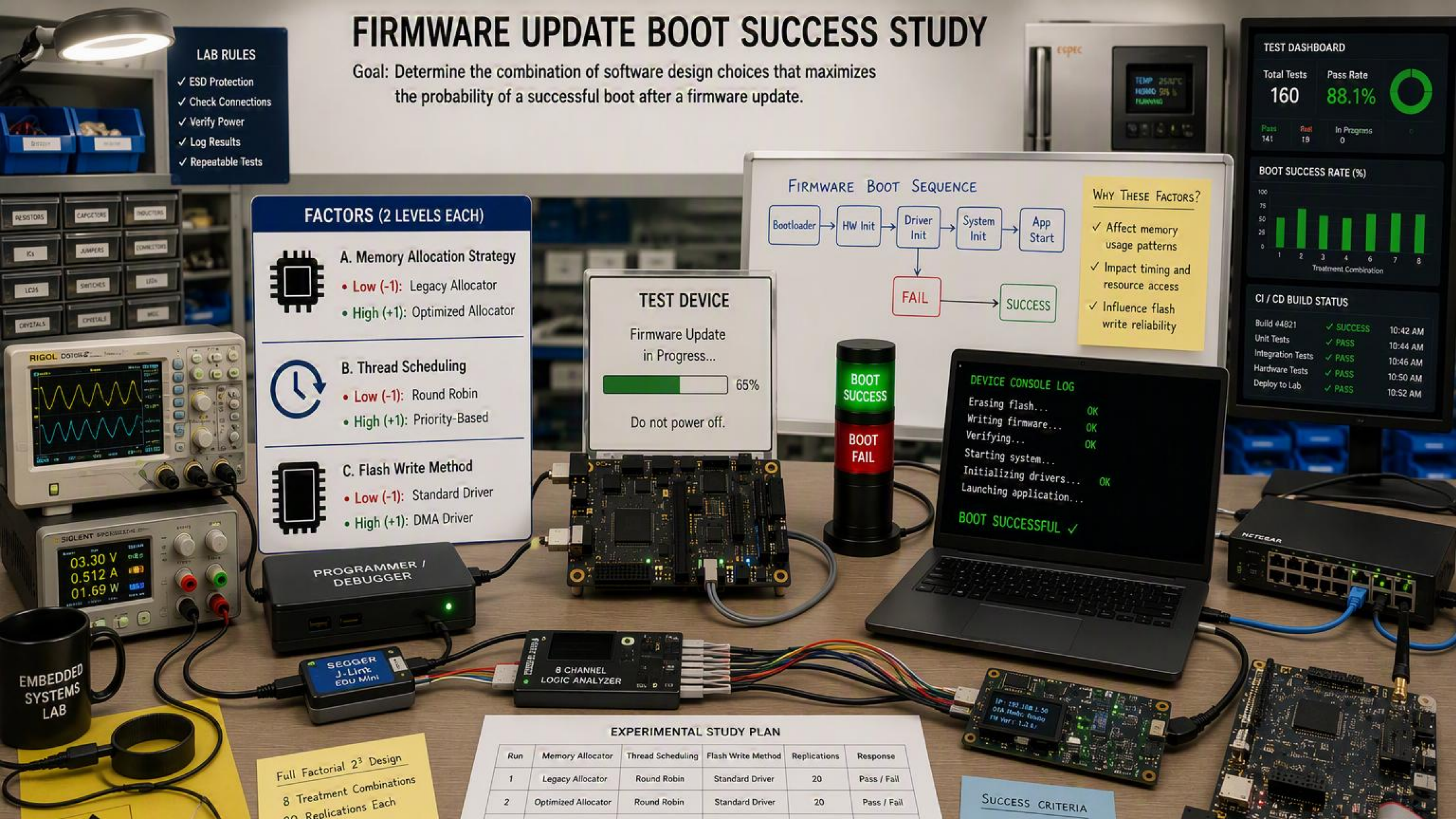
Design of Experiments (DOE) with Attribute Response

July 20, 2026

Ramon Balisnomo



FIRMWARE UPDATE BOOT SUCCESS STUDY
Goal: Determine the combination of software design choices that maximizes the probability of a successful boot after a firmware update.
LAB RULES
✓ ESD Protection
✓ Check Connections
✓ Verify Power
✓ Log Results
✓ Repeatable Tests
TEST DASHBOARD
Total Tests
160
Pass Rate
88.1%
In Progress
0
BOOT SUCCESS RATE (%)
Treatment Combination
CI / CD BUILD STATUS
Build #4821 ✓ SUCCESS 10:42 AM
Unit Tests ✓ PASS 10:44 AM
Integration Tests ✓ PASS 10:46 AM
Hardware Tests ✓ PASS 10:50 AM
Deploy to Lab ✓ PASS 10:52 AM
FACTORS (2 LEVELS EACH)
A. Memory Allocation Strategy
• Low (-1): Legacy Allocator
• High (+1): Optimized Allocator
B. Thread Scheduling
• Low (-1): Round Robin
• High (+1): Priority-Based
C. Flash Write Method
• Low (-1): Standard Driver
• High (+1): DMA Driver
TEST DEVICE
Firmware Update in Progress...
65%
Do not power off.
FIRMWARE BOOT SEQUENCE
Bootloader → HW Init → Driver Init → System Init → App Start
FAIL → SUCCESS
WHY THESE FACTORS?
✓ Affect memory usage patterns
✓ Impact timing and resource access
✓ Influence flash write reliability
BOOT SUCCESS
BOOT FAIL
DEVICE CONSOLE LOG
Erasing flash... OK
Writing firmware... OK
Verifying... OK
Starting system...
Initializing drivers... OK
Launching application...
BOOT SUCCESSFUL ✓
RIGOL
PROGRAMMER / DEBUGGER
03.30 V
0.512 A
01.69 W
SEGGER J-Link EDU Mini
8 CHANNEL LOGIC ANALYZER
NETGEAR
EMBEDDED SYSTEMS LAB
Full Factorial 2^3 Design
8 Treatment Combinations
20 Replications Each
EXPERIMENTAL STUDY PLAN
Run | Memory Allocator | Thread Scheduling | Flash Write Method | Replications | Response
1 | Legacy Allocator | Round Robin | Standard Driver | 20 | Pass / Fail
2 | Optimized Allocator | Round Robin | Standard Driver | 20 | Pass / Fail
SUCCESS CRITERIA

Case Study: Firmware Deployment Reliability

Firmware engineering would like to predict the probability of a successful firmware deployment based on key software architecture decisions. The objective is to determine how different implementation strategies affect the likelihood that an embedded device successfully boots and passes verification after a firmware update.

Response Variable: Pass (1) = successful deployment and boot; Fail (0) = deployment or boot failure.

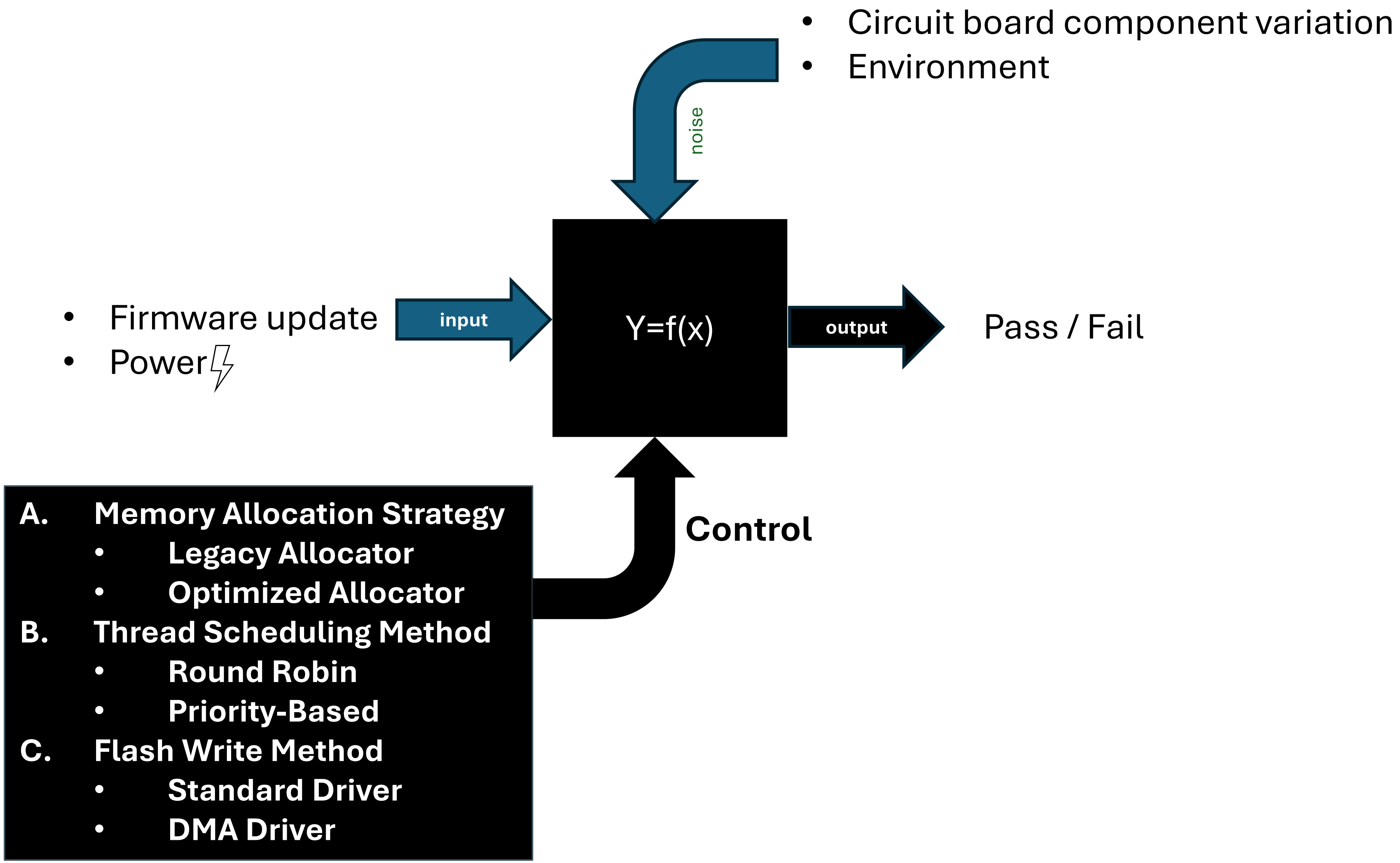
Experimental Factors

A. Memory Allocation Strategy (Legacy Allocator vs Optimized Allocator)

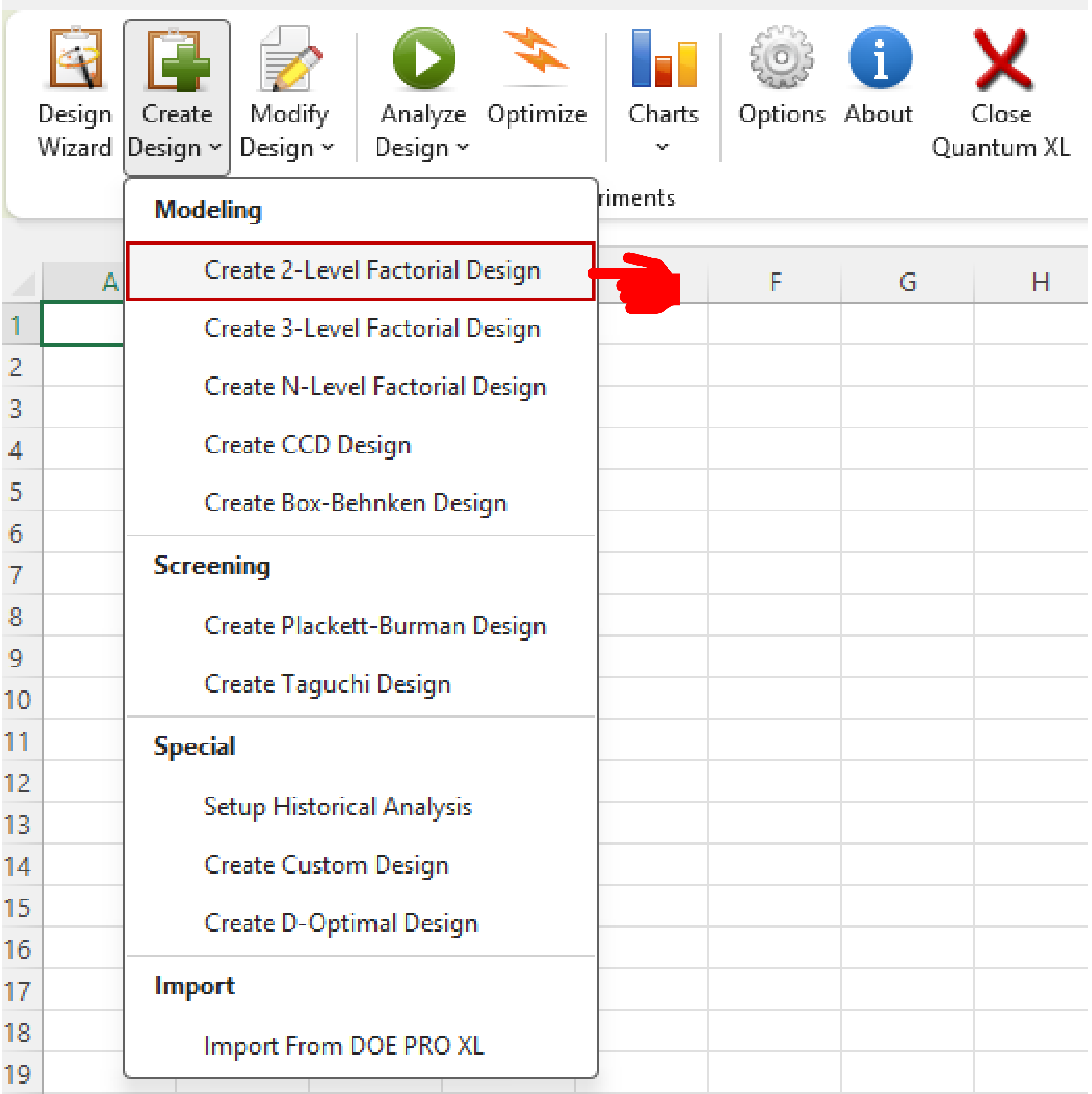
B. Thread Scheduling Method (Round Robin vs Priority-Based)

C. Flash Write Method (Standard Driver vs DMA Driver)

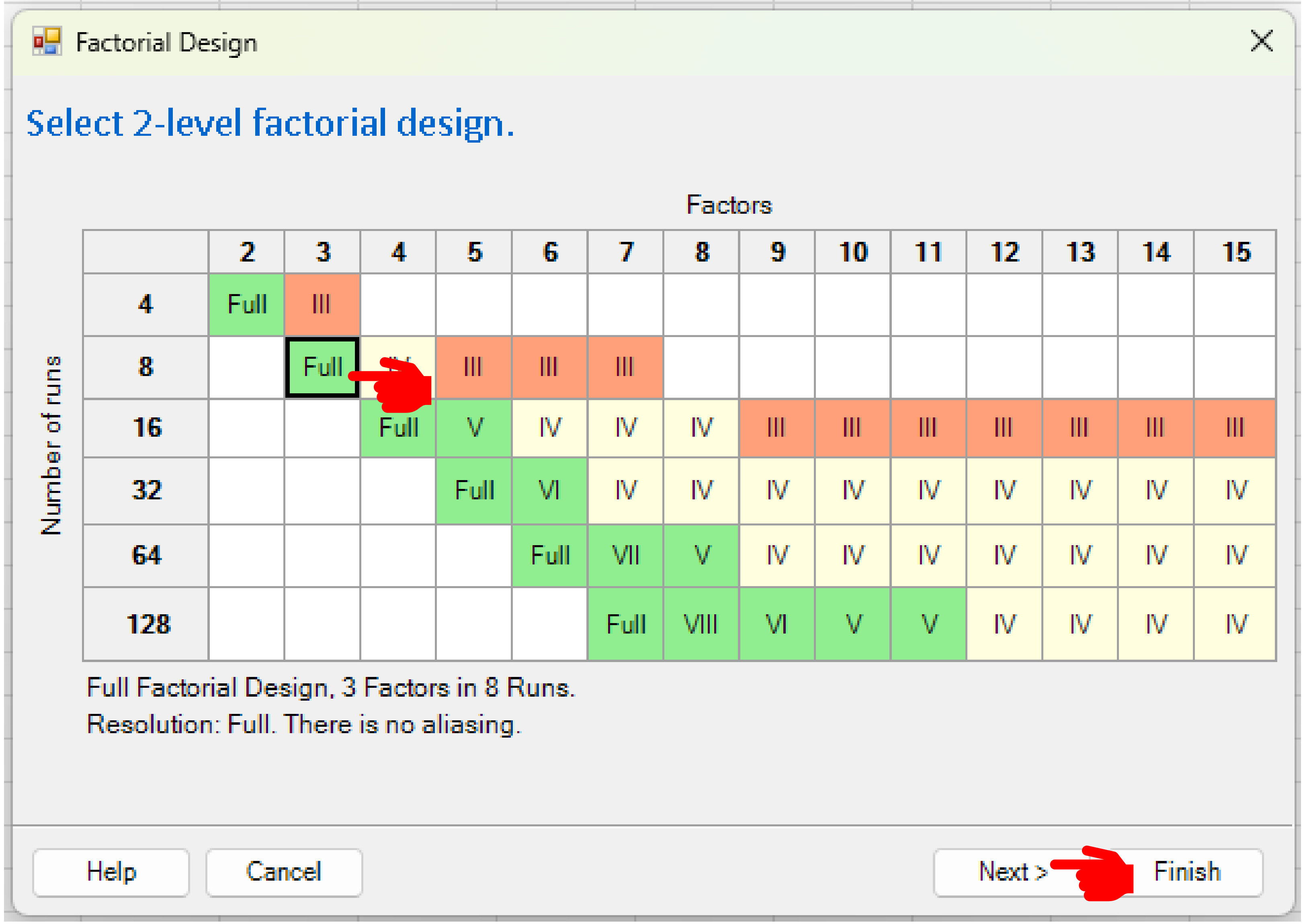
Parameter Diagram



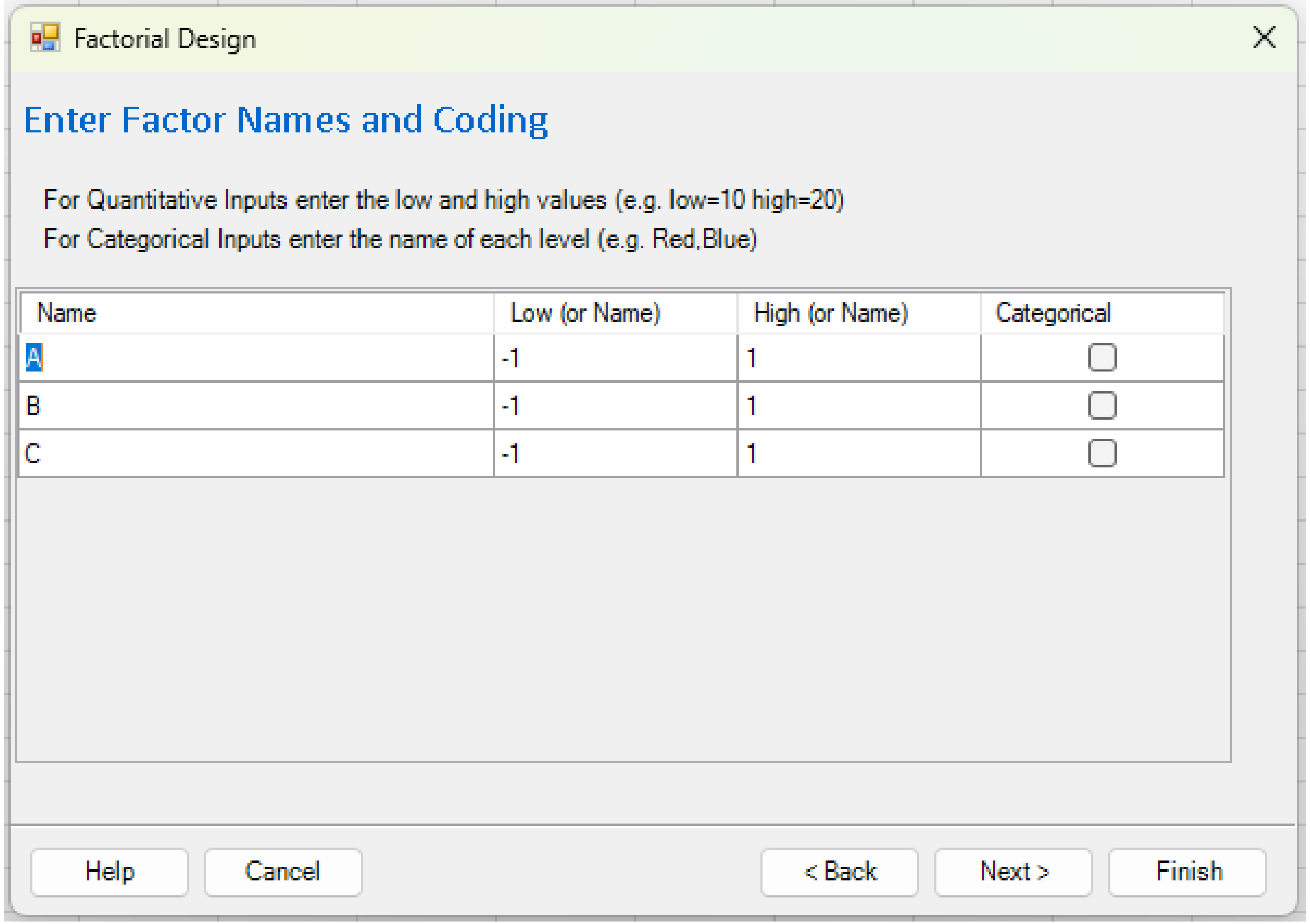
QXL DOE >Create Design >Create 2-Level Factorial Design



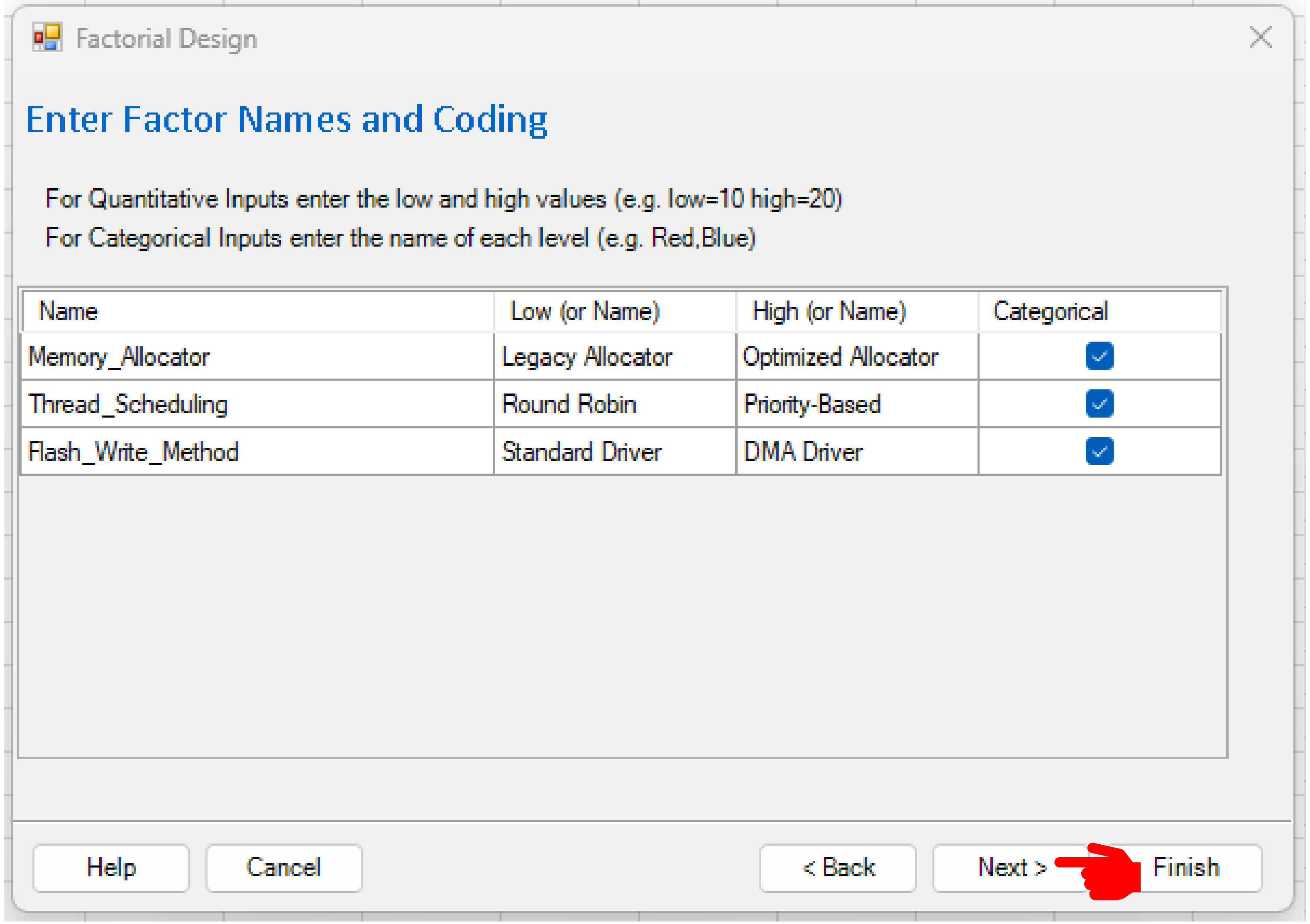
Select 3 factors with 8 number of runs



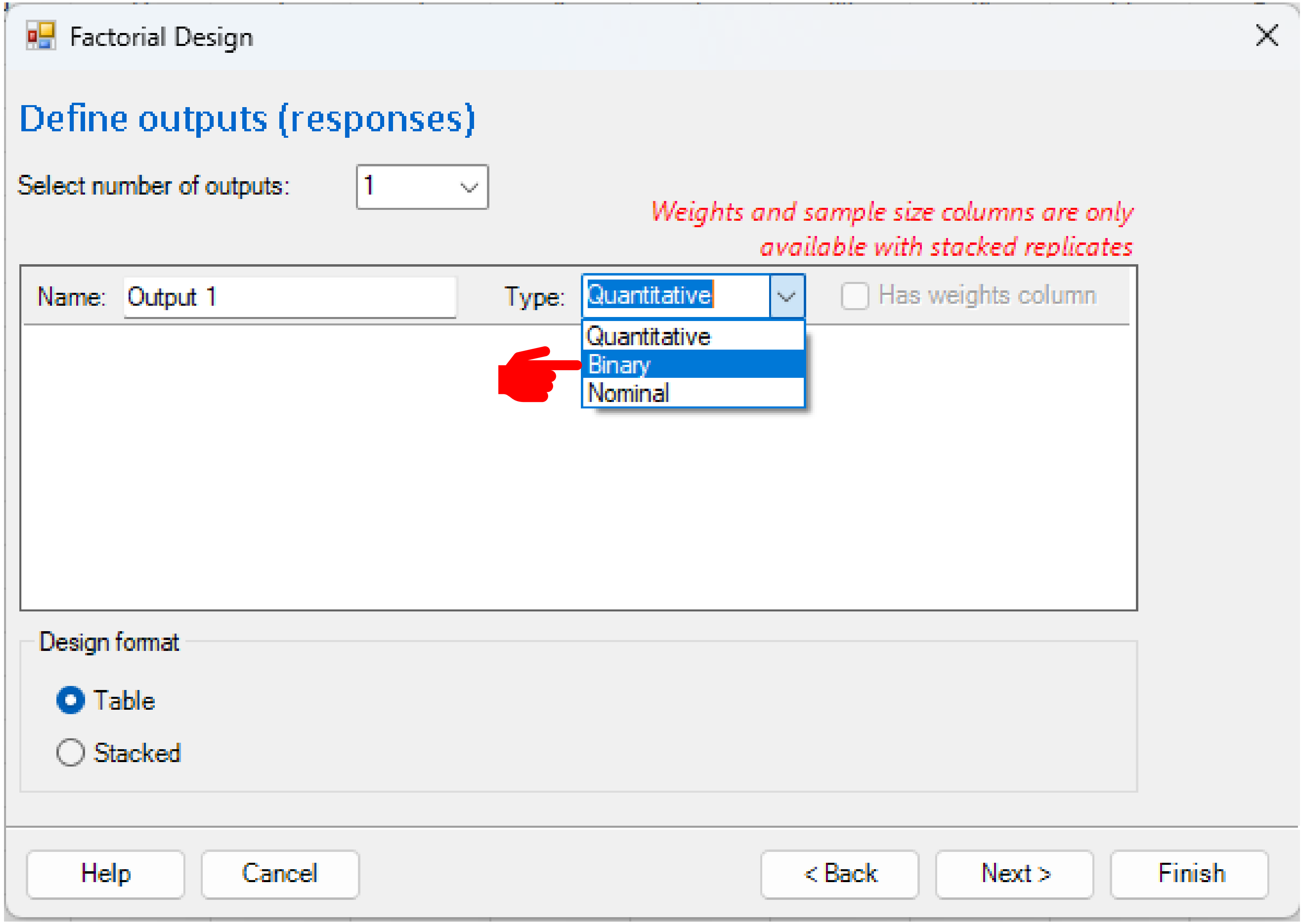
Enter factor names and high and low settings for each factor



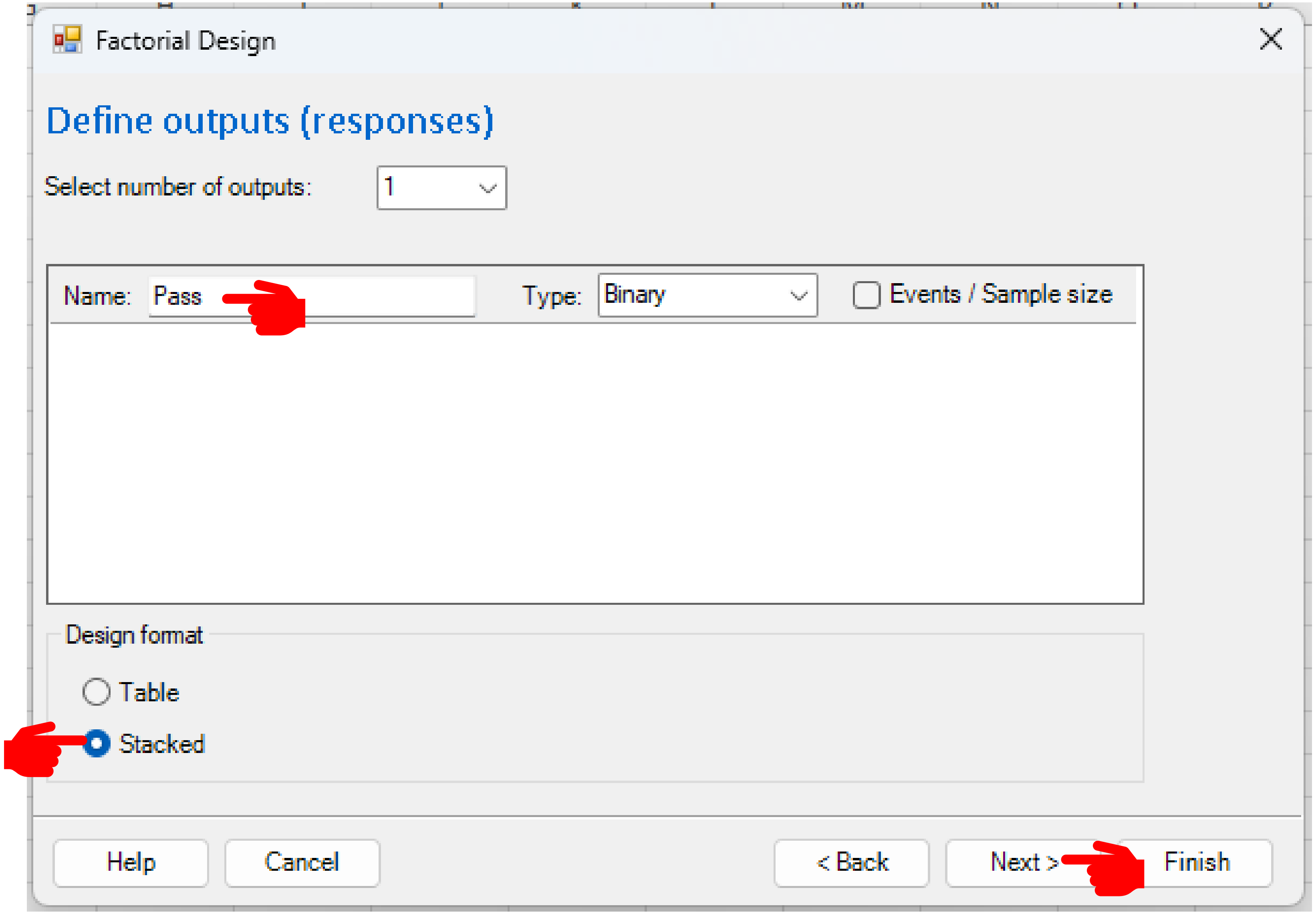
Enter factor names and high and low settings for each factor



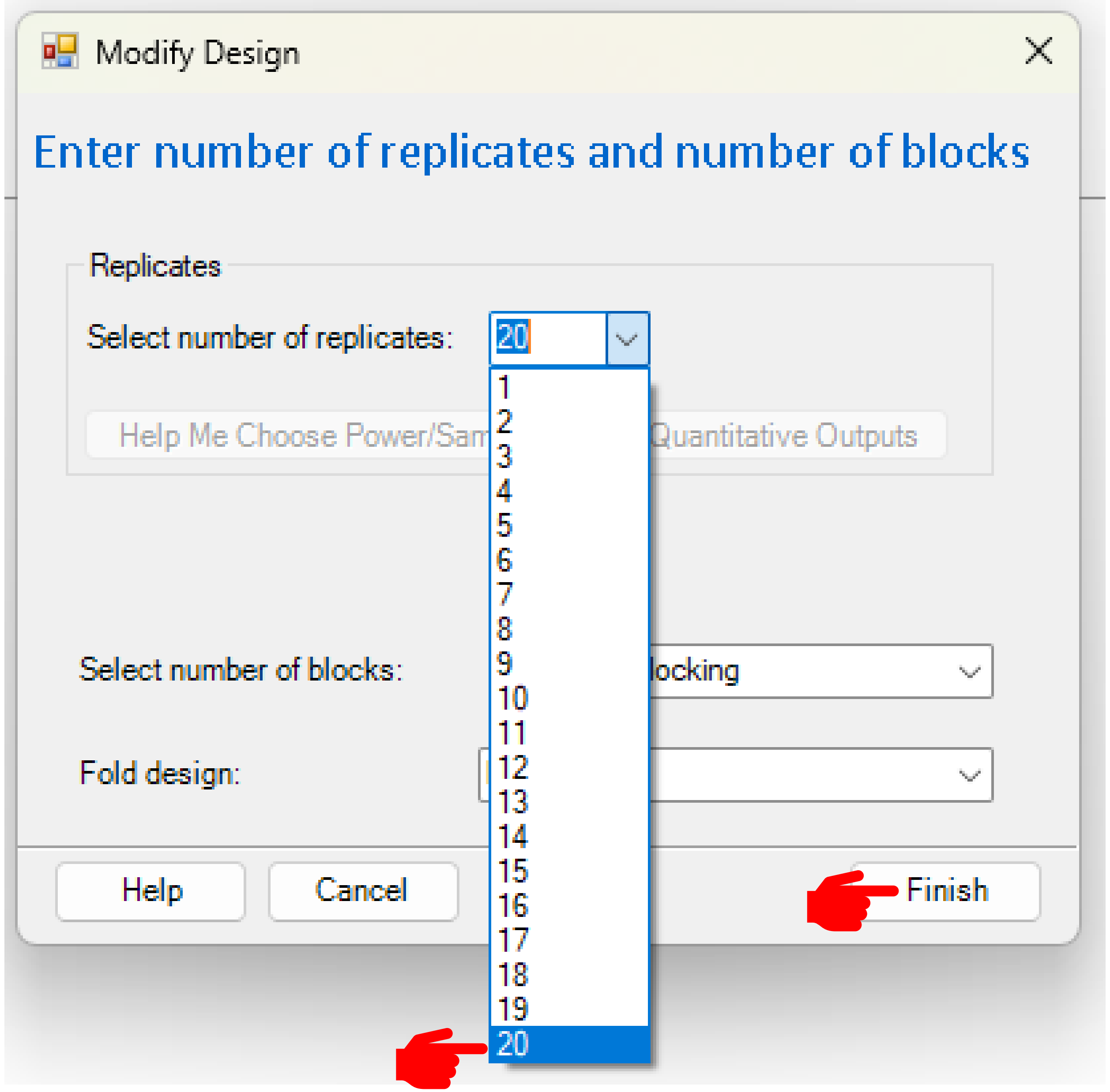
Choose binary for output type.



Complete the dialog box as shown before clicking Next.



Select 20 for number of replicates.



Why so many replications?

Treatment	Combination	Pass	Fail	Replications	Pass Rate
1	Legacy, RR, Standard	1	9	**10**	10%
2	Optimized, RR, Standard	3	7	**10**	30%
3	Legacy, Priority, Standard	3	7	**10**	30%
4	Optimized, Priority, Standard	9	1	**10**	90%
5	Legacy, RR, DMA	2	8	**10**	20%
6	Optimized, RR, DMA	8	2	**10**	80%
7	Legacy, Priority, DMA	8	2	**10**	80%
8	Optimized, Priority, DMA	10	**0**	**10**	100%

- Binary logistic regression requires both response outcomes (Pass and Fail) to be represented in the data. If a treatment combination produces all successes or all failures, the model may experience complete separation and fail to converge.
- In the original experiment with 10 replications per treatment combination, Treatment 8 produced zero failures. Because one response category was absent, the logistic regression algorithm could not reliably estimate the logit function and transfer function.
- To address this issue, additional replications were collected. Increasing the sample size improves the likelihood of observing both Pass and Fail outcomes within each treatment combination.

Why so many replications?

Treatment	Combination	Pass	Fail	Replications	Pass Rate
1	Legacy, RR, Standard	2	18	**20**	10%
2	Optimized, RR, Standard	6	14	**20**	30%
3	Legacy, Priority, Standard	6	14	**20**	30%
4	Optimized, Priority, Standard	18	2	**20**	90%
5	Legacy, RR, DMA	4	16	**20**	20%
6	Optimized, RR, DMA	16	4	**20**	80%
7	Legacy, Priority, DMA	16	4	**20**	80%
8	Optimized, Priority, DMA	19	1	**20**	95%

- After increasing to 20 replications per treatment combination, Treatment 8 produced at least one failure. This eliminated the zero-count condition and allowed the logistic regression model to converge successfully.

- Because every treatment combination now contains both Pass and Fail observations, the binary logistic regression model can estimate the coefficients and generate a valid transfer function for predicting the probability of successful firmware deployment.

- As a practical guideline, logistic regression models generally require an adequate number of observations for each model term. For a full factorial experiment with three factors and all interaction terms, a target of 20 observations per treatment combination provides a more stable and reliable model. In this example, a 2^3 full factorial design with 20 replications per treatment yields 160 total observations, which is a reasonable sample size for estimating main effects and interactions.

Quantum XL

Full Factorial Design

3 Factors in 8 Runs

Number of replicates: 20

enter 1 (success)

edit resp

	A	B	C		Pass
				enter event value (optional) -->	Event: 1
				<-- reference level	
Run	Memory_Allocator	Thread_Scheduling	Flash_Write_Method	*<-- enter factor names*	Data
145	Legacy Allocator	Round Robin	Standard Driver	Replicate #19	
146	Legacy Allocator	Round Robin	DMA Driver		
147	Legacy Allocator	Priority-Based	Standard Driver		
148	Legacy Allocator	Priority-Based	DMA Driver		
149	Optimized Allocator	Round Robin	Standard Driver		
150	Optimized Allocator	Round Robin	DMA Driver		
151	Optimized Allocator	Priority-Based	Standard Driver		
152	Optimized Allocator	Priority-Based	DMA Driver		
153	Legacy Allocator	Round Robin	Standard Driver	Replicate #20	
154	Legacy Allocator	Round Robin	DMA Driver		
155	Legacy Allocator	Priority-Based	Standard Driver		
156	Legacy Allocator	Priority-Based	DMA Driver		
157	Optimized Allocator	Round Robin	Standard Driver		
158	Optimized Allocator	Round Robin	DMA Driver		
159	Optimized Allocator	Priority-Based	Standard Driver		
160	Optimized Allocator	Priority-Based	DMA Driver		

Quantum XL

Full Factorial Design

3 Factors in 8 Runs

Number of replicates: 20

edit response name (optional) --> **Pass**

enter event value (optional) --> Event: **1**

	A	B	C	
				<-- reference level
Run	**Memory_Allocator**	**Thread_Scheduling**	**Flash_Write_Method**	**Data**
145	Legacy Allocator	Round Robin	Standard Driver	0
146	Legacy Allocator	Round Robin	DMA Driver	1
147	Legacy Allocator	Priority-Based	Standard Driver	1
148	Legacy Allocator	Priority-Based	DMA Driver	0
149	Optimized Allocator	Round Robin	Standard Driver	1
150	Optimized Allocator	Round Robin	[illegible]	1
151	Optimized Allocator	Priority-Based	[illegible]	0
152	Optimized Allocator	Priority-Based	[illegible]	1
153	Legacy Allocator	Round Robin	[illegible]	0
154	Legacy Allocator	Round Robin	DMA Driver	1
155	Legacy Allocator	Priority-Based	Standard Driver	1
156	Legacy Allocator	Priority-Based	DMA Driver	0
157	Optimized Allocator	Round Robin	Standard Driver	1
158	Optimized Allocator	Round Robin	DMA Driver	1
159	Optimized Allocator	Priority-Based	Standard Driver	1
160	Optimized Allocator	Priority-Based	DMA Driver	0

<-- enter factor names

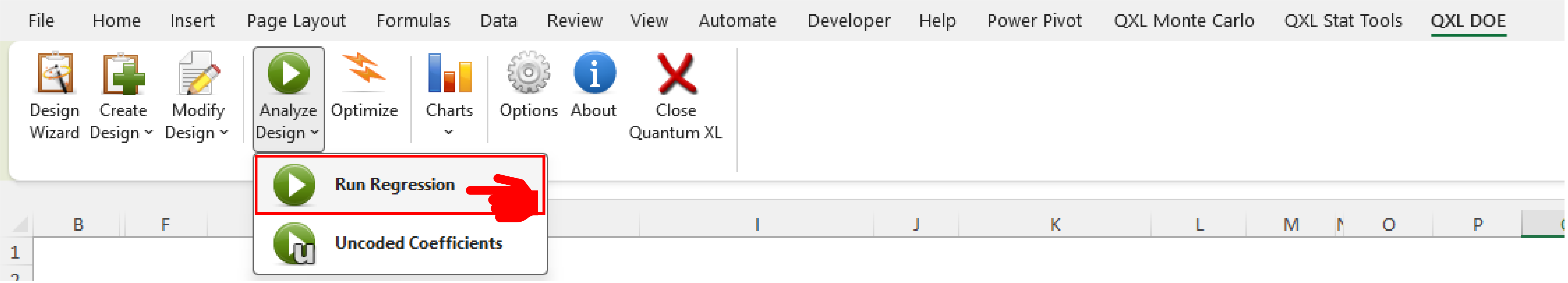
Replicate #19

run experiments and record results

DOE Data in Excel

Open Embedded File

QXL DOE >Analyze Design >Run Regression



The transfer function will take this form:

$$\mathrm{logit}(p) = \ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_A A + \beta_B B + \beta_C C + \beta_{AB} AB + \beta_{AC} AC + \beta_{BC} BC$$

Degrees of freedom (DF) should not be zero. There are too many terms in the model. The P-value for the Goodness of Fit Tests cannot be calculated.

Goodness of Fit Test

	Chi-Sq	DF	P
Pearson	0.0	0	NA
Deviance	0.0	0	NA

Association

	Number	Percent
Concordant	4,804	75.06%
Discordant	724	11.31%
Ties	872	13.63%
Total	6,400	100.00%

Summary Measures

	Value
Somers' D	0.6375
Goodman-Kruskal Gamma	0.7381
Kendall's Tau	0.3208

Remove the 3-way interaction ABC. Run regression model again.

There are no warnings for this model. Click here to see DOE Advisor.

Pass

Binary Logistic Output - Converged in 7 iterations.

Name	Factor	Coeff	SE	Z	P	Odds Ratio	95% CI Lower	95% CI Upper	In Model
	Const	-2.1972	0.7454	-2.9479	0.003	0.1111	0.0258	0.4789	
Memor	**Memory_Allocator (A)**	1.3499	0.8909	1.5153	0.130	3.8571	0.6729	22.109	☑
Thread	**Thread_Scheduling (B)**	1.3499	0.8909	1.5153	0.130	3.8571	0.6729	22.109	☑
Flash_\	**Flash_Write_Method (C)**	0.8109	0.9317	0.8704	0.384	2.25	0.3623	13.971	☑
Memor	**AB**	0.8837	1.1594	0.7622	0.446	2.4198	0.2494	23.478	☑
Memor	**AC**	0.8837	1.1594	0.7622	0.446	2.4198	0.2494	23.478	☑
Thread	**BC**	0.8837	1.1594	0.7622	0.446	2.4198	0.2494	23.478	☑
Memor	**ABC**	-1.7673	1.6397	-1.0779	0.281	0.1708	0.0069	4.2[illegible]	☐

Remove 2-way Interactions because of the high P-values. Run regression a third time.

There are no warnings for this model. Click here to see DOE Advisor.

Pass

Binary Logistic Output - Converged in 9 iterations.

Name	Factor	Coeff	SE	Z	P	Odds Ratio	95% CI Lower	95% CI Upper	In Model
	Const	-2.6062	0.7613	-3.4234	0.001	0.0738	0.0166	0.3282	
Memory_Allocator (ref='Legacy Allocator')	**Memory_Allocator (A)**	1.9037	0.8197	2.3223	0.020	6.7107	1.3459	33.46	☑
Thread_Scheduling (ref='Round Robin')	**Thread_Scheduling (B)**	1.9037	0.8197	2.3223	0.020	6.7107	1.3459	33.46	☑
Flash_Write_Method (ref='Standard Driver')	**Flash_Write_Method (C)**	1.405	0.8304	1.692	0.091	4.0755	0.8005	20.75	☑
Memory_Allocator*Thread_Scheduling	**AB**	0.0	0.87	0.0	1.000	1.0	0.1817	5.5021	☐
Memory_Allocator*Flash_Write_Method	**AC**	0.0	0.87	0.0	1.000	1.0	0.1817	5[illegible]	☐
Thread_Scheduling*Flash_Write_Method	**BC**	0.0	0.87	0.0	1.000	1.0	0.1817	5.5021	☐

Introduction to Odds Ratio

$$log\left(\frac{p}{1-p}\right) = log(odds)$$

- The odds ratio (OR) is often the most useful output from a logistic regression because it translates statistical coefficients into something engineers and managers can understand: "How much does changing a factor increase or decrease the odds of success?"
- Odds ratio > 1 indicates the odds of the event are increased by the variable. Odds ratio < 1 indicates the odds of the event are decreased by the variable.
- For example, changing the flash write method from the low setting of standard driver to the high setting of DMA drive increases successful deployment odds by approximately 4x.

Name	Factor	P	Odds Ratio	95% CI Lower	95% CI Upper	In Model
	Const	0.000	0.0738	0.0291	0.1872	
Memory_Allocator (ref='Legacy Allocator')	Memory_Allocator (A)	0.000	6.7107	3.001	15.006	☑
Thread_Scheduling (ref='Round Robin')	Thread_Scheduling (B)	0.000	6.7107	3.001	15.006	☑
Flash_Write_Method (ref='Standard Driver')	Flash_Write_Method (C)	0.000	4.0755	1.8681	8.891	☑

What does the **G P-value** tell us?

- It tells us whether the factors in our experiment collectively help predict firmware deployment success. A small **G P-value** means the software architecture choices have a real effect on reliability; a large **G P-value** means the model performs no better than random guessing based on the overall average success rate.

- If the **G P-value < 0.05**, we are at least 95% confident that the slopes are not at all equal to zero (the model is significant for prediction).

- Think of the **G P-value** as the logistic-regression equivalent of the overall ANOVA Model P-value.

G P-Value tells us the transfer function is significant.

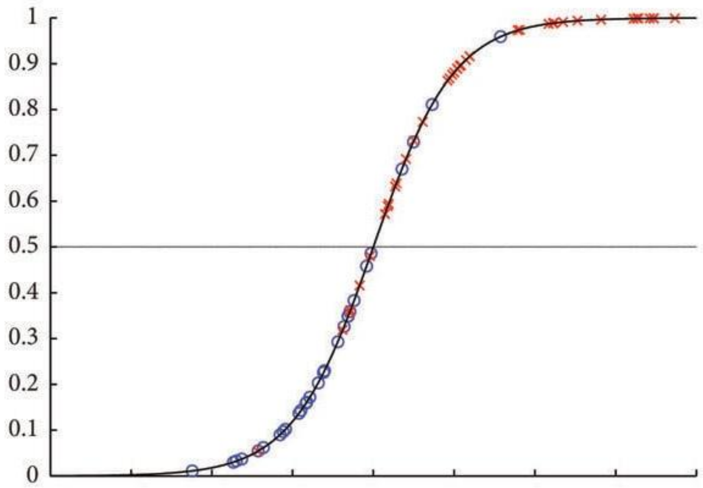
There are no warnings for this model. Click here to see DOE Advisor.

Pass

Binary Logistic Output - Converged in 6 iterations.

Name	Factor	Coeff	SE	Z	P	Odds Ratio	95% CI Lower	95% CI Upper	In Model
	Const	-2.6062	0.4749	-5.4879	0.000	0.0738	0.0291	0.1872	
Memory_Allocator (ref='Legacy Allocator')	Memory_Allocator (A)	1.9037	0.4106	4.6365	0.000	6.7107	3.001	15.006	☑
Thread_Scheduling (ref='Round Robin')	Thread_Scheduling (B)	1.9037	0.4106	4.6365	0.000	6.7107	3.001	15.006	☑
Flash_Write_Method (ref='Standard Driver')	Flash_Write_Method (C)	1.405	0.398	3.5302	0.000	4.0755	1.8681	8.891	☑

LogLikelihood	-82.452
RSquaredU	0.2565
AIC	172.904
BIC	185.204
RMSE	0.4105
G Stat	56.903
G df	3
G P-Value	0.000



For Illustrative Purpose Only

Goodness of Fit (GOF) Test for Binary Logistic Regression

GOF Test	Question Answered and Description
Pearson	Are observed counts close to predicted counts? It compares the predicted pass rate to the actual pass rate. $$X^2 = \sum \frac{(Observed - Expected)^2}{Expected}$$
Deviance	How close is the model to a perfect model? This test is based on likelihoods. A perfect model would exactly predict every observed proportion. $$D = 2 \times (\text{Log Likelihood of Perfect Model} - \text{Log Likelihood of Current Model})$$

- If both P-values > 0.05, it means the model is both statistically significant and practically representative of the data.
- The Deviance GOF is preferred over the Pearson because it is directly tied to the likelihood function used to estimate logistic regression coefficients. Many statisticians consider the Deviance the primary GOF metric.

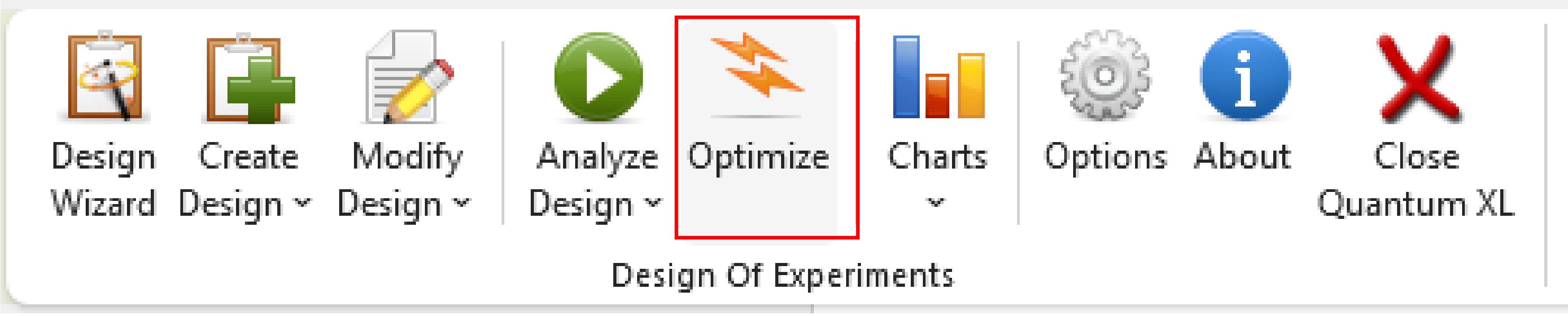
The P-values for Pearson and Deviance Good of Fit Test indicate the model is both statistically significant and practically representative of the data.

LogLikelihood	-82.452
RSquaredU	0.2565
AIC	172.904
BIC	185.204
RMSE	0.4105
G Stat	56.903
G df	3
G P-Value	0.000

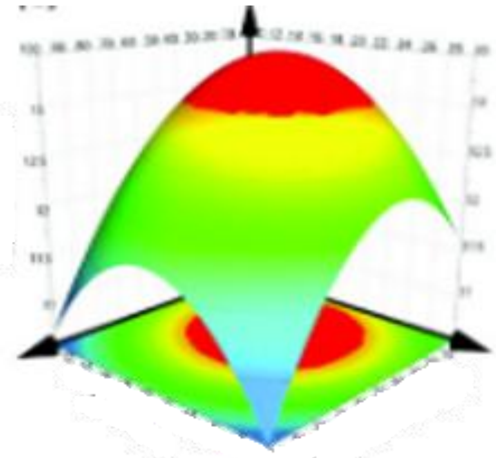
Goodness of Fit Test

	Chi-Sq	DF	P
Pearson	1.1833	4	0.881
Deviance	1.1266	4	0.890

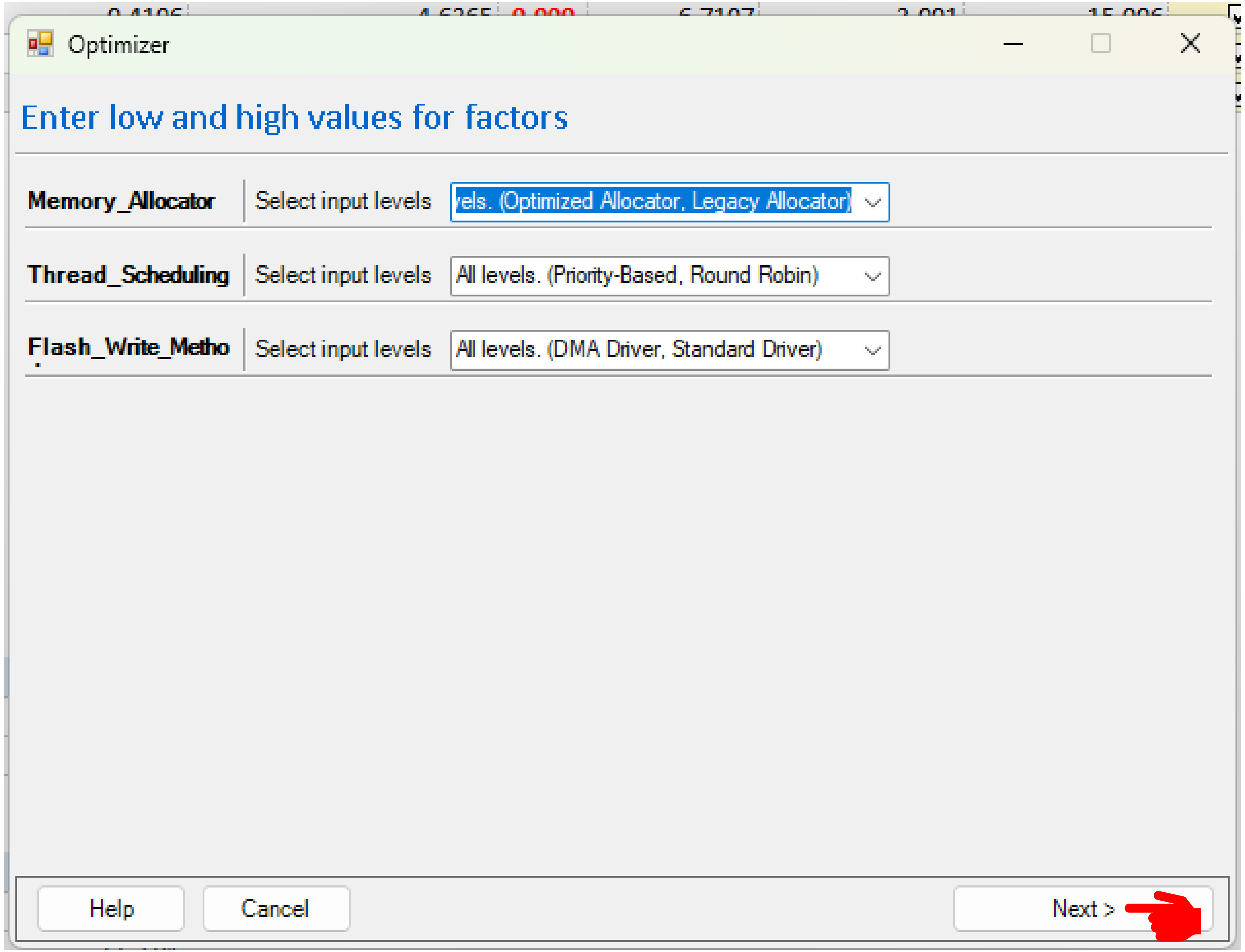
QXL DOE >Optimize

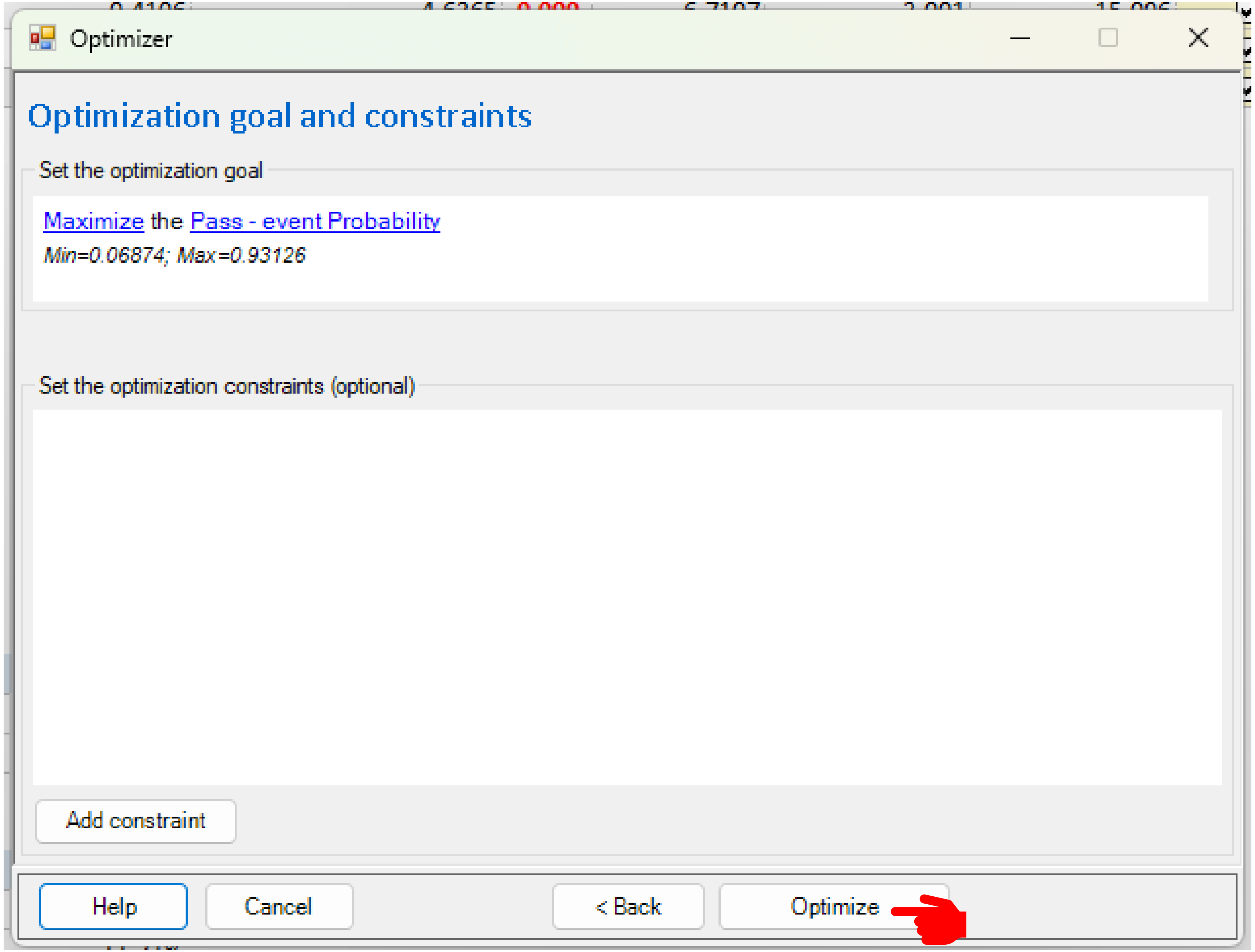


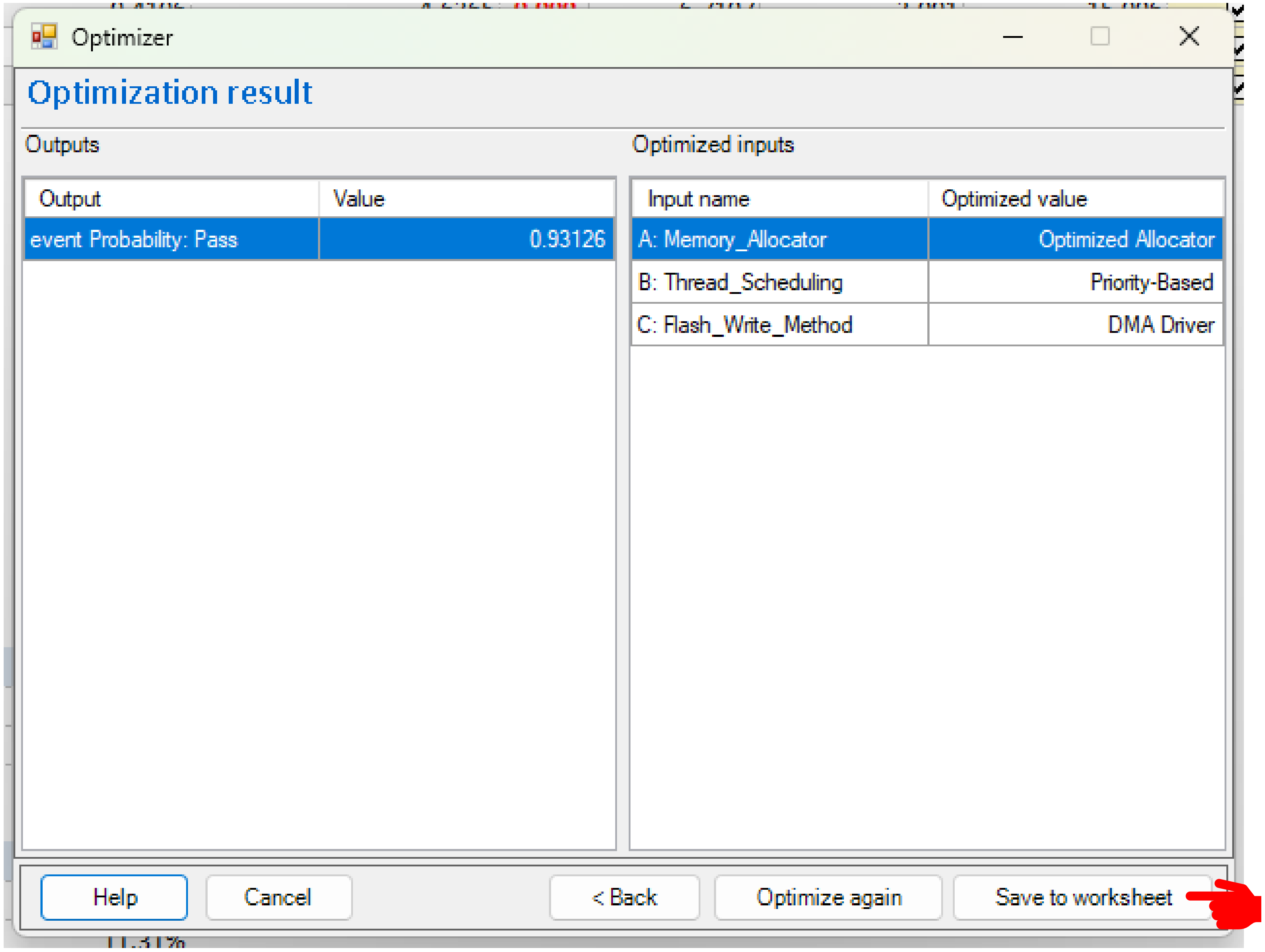
Find the set points that will yield the highest success rate.



For Illustrative Purpose Only


Optimizer
Enter low and high values for factors
Memory_Allocator
Select input levels
vels. (Optimized Allocator, Legacy Allocator)
Thread_Scheduling
Select input levels
All levels. (Priority-Based, Round Robin)
Flash_Write_Metho
Select input levels
All levels. (DMA Driver, Standard Driver)
Help
Cancel
Next >


Optimizer
Optimization goal and constraints
Set the optimization goal
Maximize the Pass - event Probability
Min=0.06874; Max=0.93126
Set the optimization constraints (optional)
Add constraint
Help
Cancel
< Back
Optimize


Optimizer
Optimization result
Outputs
Optimized inputs
Output
Value
event Probability: Pass
0.93126
Input name
Optimized value
A: Memory_Allocator
Optimized Allocator
B: Thread_Scheduling
Priority-Based
C: Flash_Write_Method
DMA Driver
Help
Cancel
< Back
Optimize again
Save to worksheet

Below are the settings for the highest probability of success (93%).

Quantum XL Regression Analysis

Design sheet: DOE Design

Regression in coded units

Factors				
Name	Factor	Range		Set point
		Low	High	
Memory_Allocator	A	Legacy Allocator	Optimized Allocator	Optimized Allocator
Thread_Scheduling	B	Round Robin	Priority-Based	Priority-Based
Flash_Write_Method	C	Standard Driver	DMA Driver	DMA Driver

Prediction	
Pass	
	Probability
0	7%
1 (Event)	93%

Key Takeaways

1) **Binary Logistic Regression Requirements**
   - Use a minimum of **20 observations per model term** to obtain stable coefficient estimates and reliable prediction performance.
   - Ensure that every treatment combination contains at least one success and one failure to avoid complete separation and convergence issues.
2) **Goodness-of-Fit Evaluation**
   - The **Pearson** and **Deviance Goodness-of-Fit Tests** assess whether the logistic regression model adequately represents the observed data.
   - Because the null hypothesis assumes the model fits the data, larger P-values (> 0.05) indicate no evidence of lack of fit and are therefore desirable.
3) **Transfer Function Development.** The fitted binary logistic regression model generates a transfer function that predicts the probability of a successful firmware deployment as a function of the software architecture factors and their interactions.