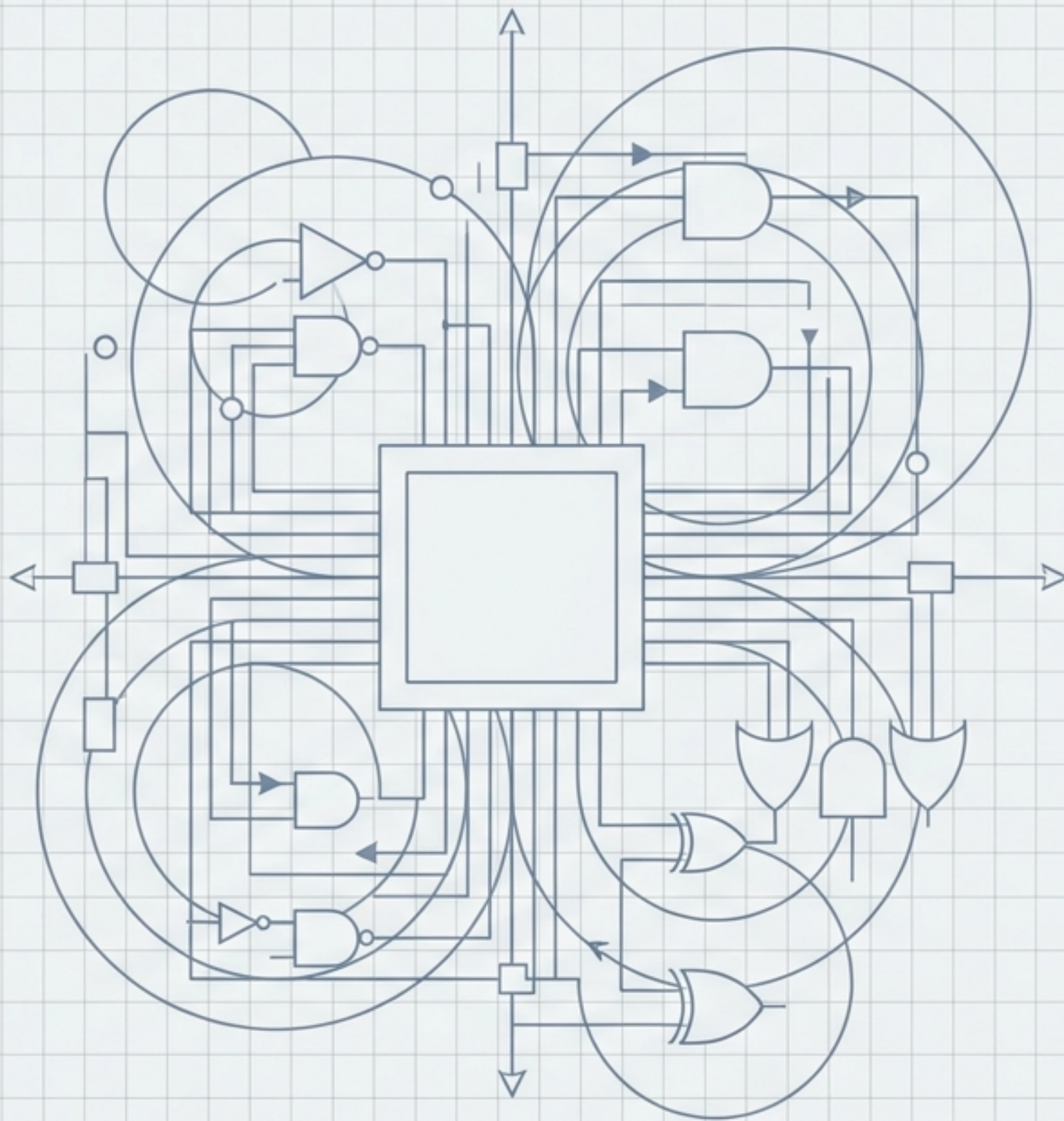


応用情報技術者試験 徹底攻略：OSタスク制御の 「思考回路」

平成21年 春期 午後 問7 設問3 –
答えを暗記するのではなく、システム仕様から論理を導き出すプロセス

THE BLUEPRINT & LOGIC GATE



MISSION BRIEFING: 本日のターゲット

STATUS: CLASSIFIED // ACCESS GRANTED



TARGETS ACQUIRED

TARGET 1: 設問3(1)

クロック制御タスクの優先度はどのように設定すべきか？

TARGET 2: 設問3(2)

クロック制御タスクが実行状態にあるとき、ほかのタスクが取り得る状態は？



WARNING: 解答非公開プロトコル

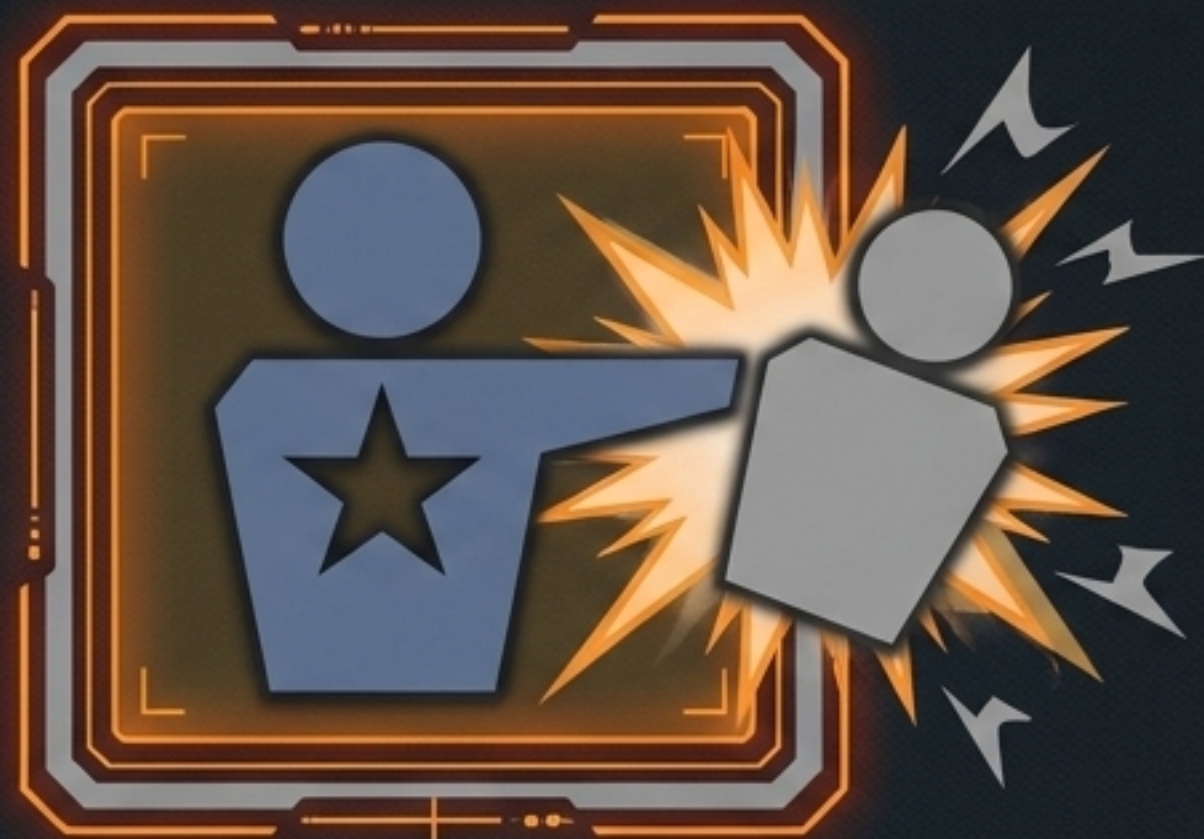
この資料に「**最終的な解答**」は記載されていません。

提供するのは、仕様書（問題文）から
正解を自力で導き出すための
「完璧な道筋（ヒント）」のみです。

Rule 01: リアルタイムOSの絶対的階級

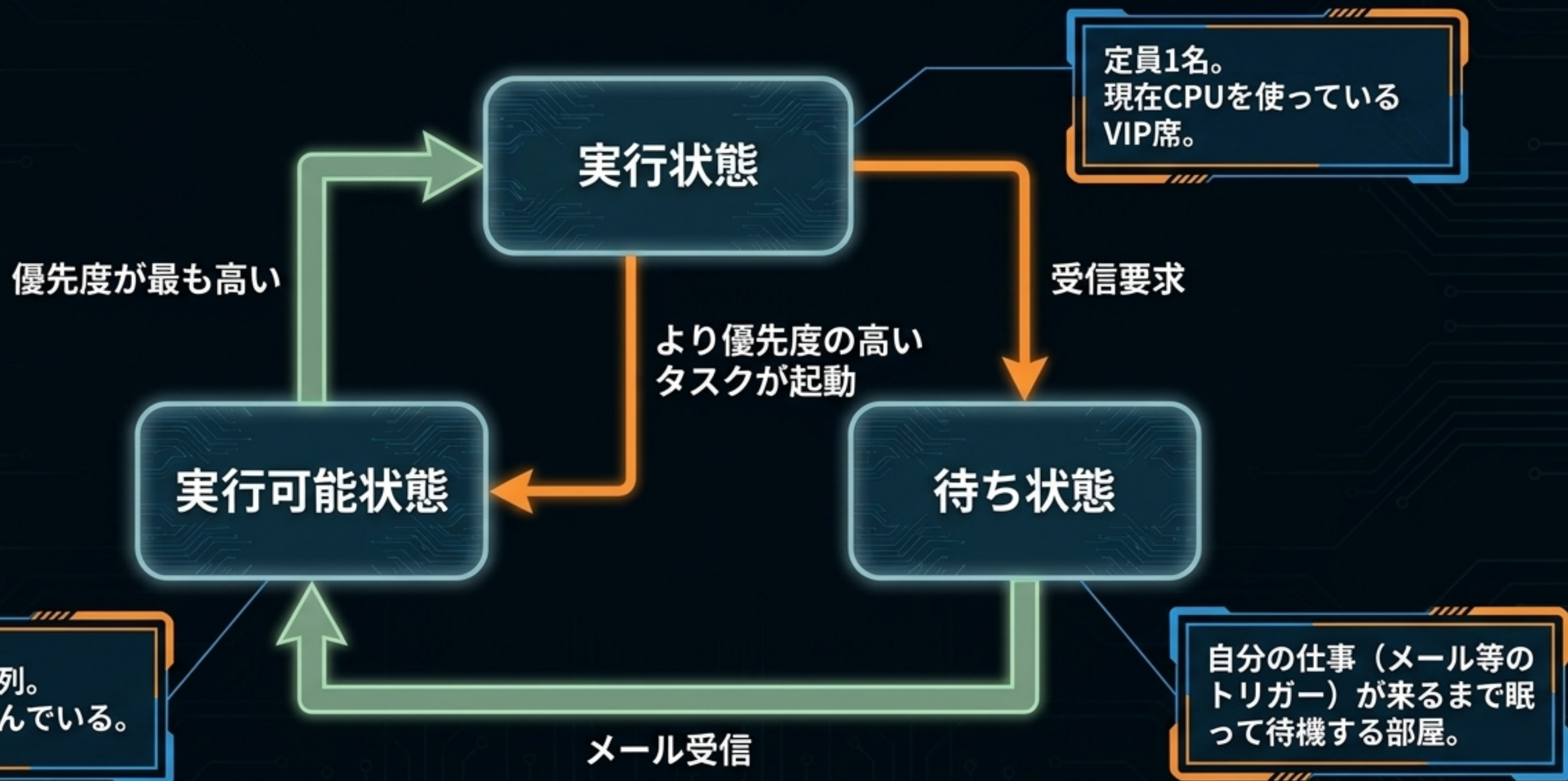
Concept: プリエンプション (Preemption)

CPU
(実行状態)



実行中のタスクより【優先度の高い】タスクが現れた瞬間、問答無用でCPUの実行権が奪われる。
システム内では、優先度こそがすべてを支配する絶対ルールである。

Rule 02: タスクの「状態遷移」マップ



Actor Profiling: 登場人物の行動原理

	 一般のタスク 	 クロック制御タスク 
役割	メールの解析と業務処理	発振停止とアイドルモードへの移行
目的	「仕事」を完了させること	MPUを「休ませる」こと
活動サイクル	メールが来たら働き、終われば待機	ひたすら「休ませる指示」をループし続ける

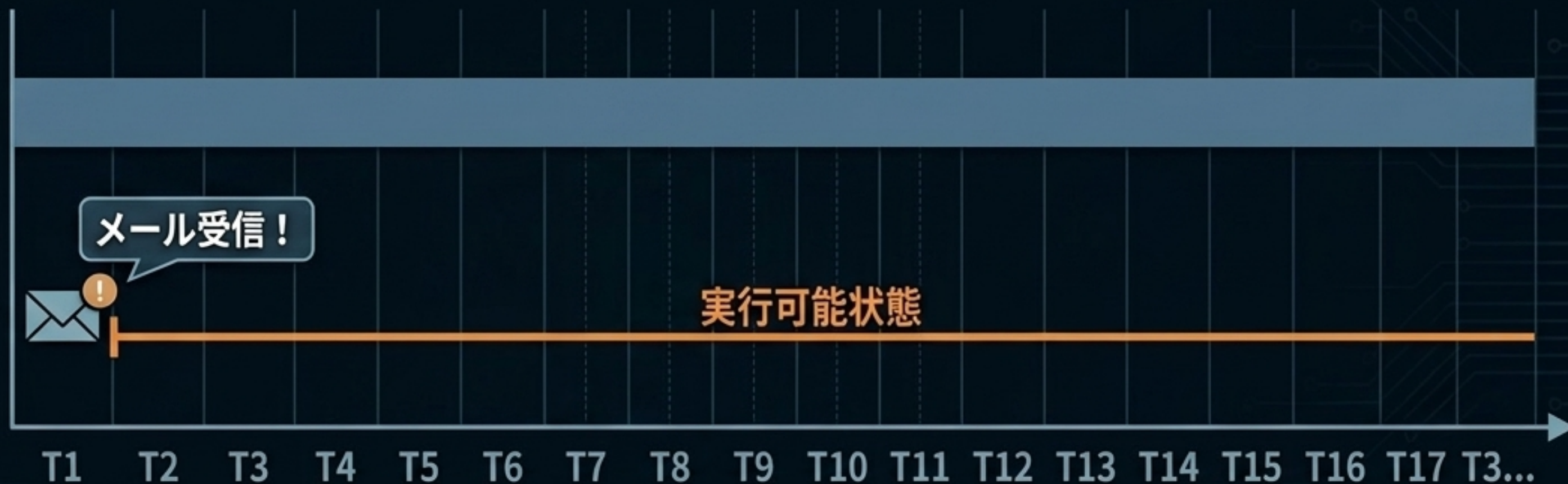
Insight: 一般タスクはシステムの「**主役**」。クロック制御タスクは、主役がいない時に電源を落とす「**裏方**」である。

Q3(1) Simulation: もし「裏方」の優先度が高かったら？

仮想シナリオ：クロック制御タスクの優先度を「高く」設定した世界線

クロック制御タスク

一般タスク



1. クロック制御タスクが常に最優先で動き続ける。
2. MPUは即座にアイドルモード（省電力）へ移行。
3. メールが届いても、一般タスクはCPUを奪えず、システムはフリーズ状態に陥る。

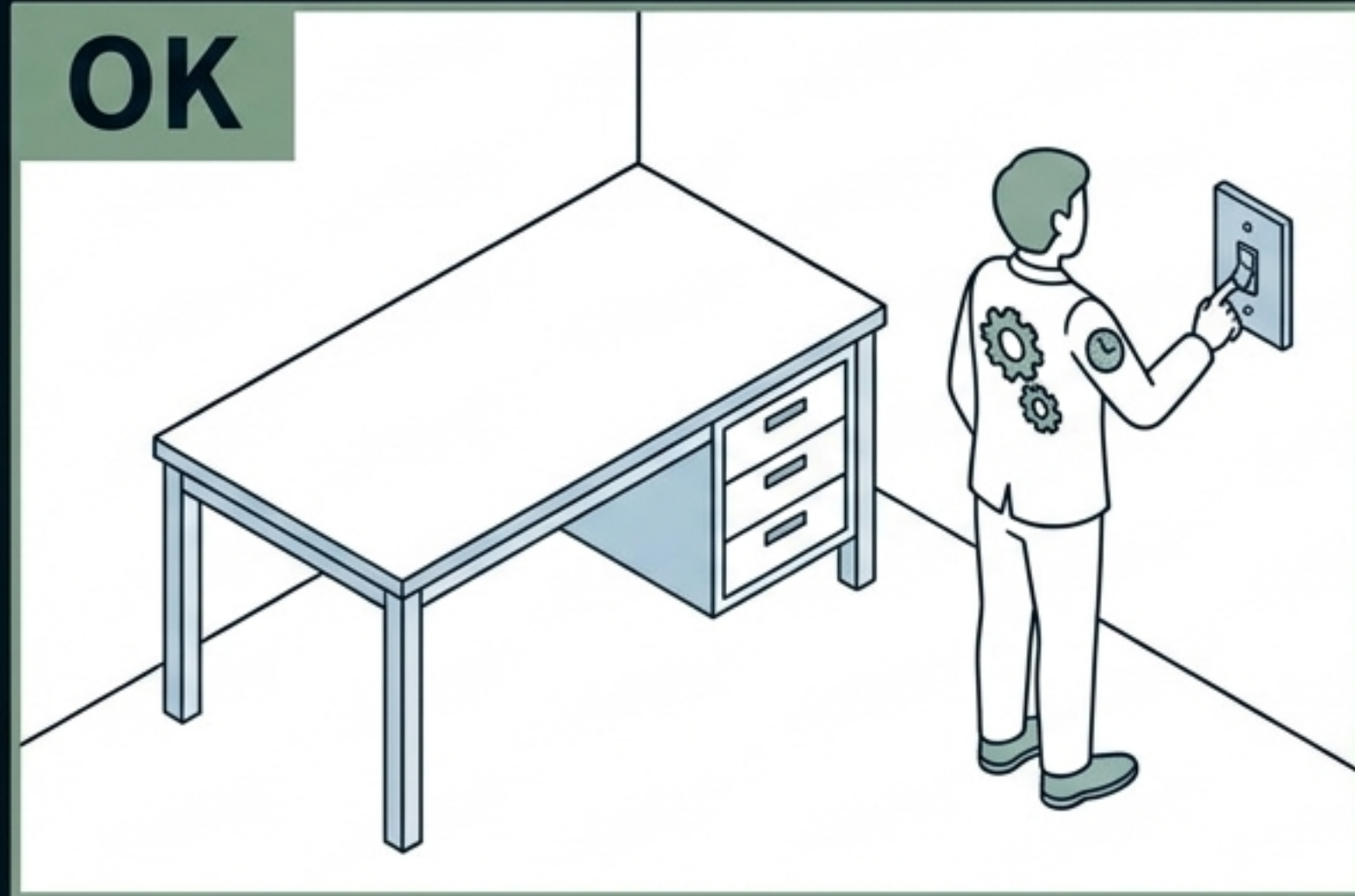
Q3(1) Core Question: 「正しい消灯タイミング」とは？

NG



処理すべき仕事（タスク）が残っているのに休ませるのは**本末転倒**。

OK



省電力モード（アイドル）に入るべき唯一のタイミングは、「システム内に**処理すべき仕事**が一切なくなった時」である。

Q3(1) Deduction: elision: 優先度の論理的帰結

クロック制御タスクの目的は、MPUをアイドル状態（省電力）にすること。

ただし、他のタスク（仕事）の実行を妨げてはならない。

したがって、クロック制御タスクが実行権を得てよいのは、
「他に実行すべきタスクが一つも無い時」だけである。

つまり、クロック制御タスクの優先度は、ほかのタスクと比べてどのように設定すべきか？

➡ 【 ? 】

Q3(2) Setup: 状況のプロファイリング

前提条件：「クロック制御タスクが『**実行状態**』にあるとき」

待ち状態

実行可能状態

実行状態

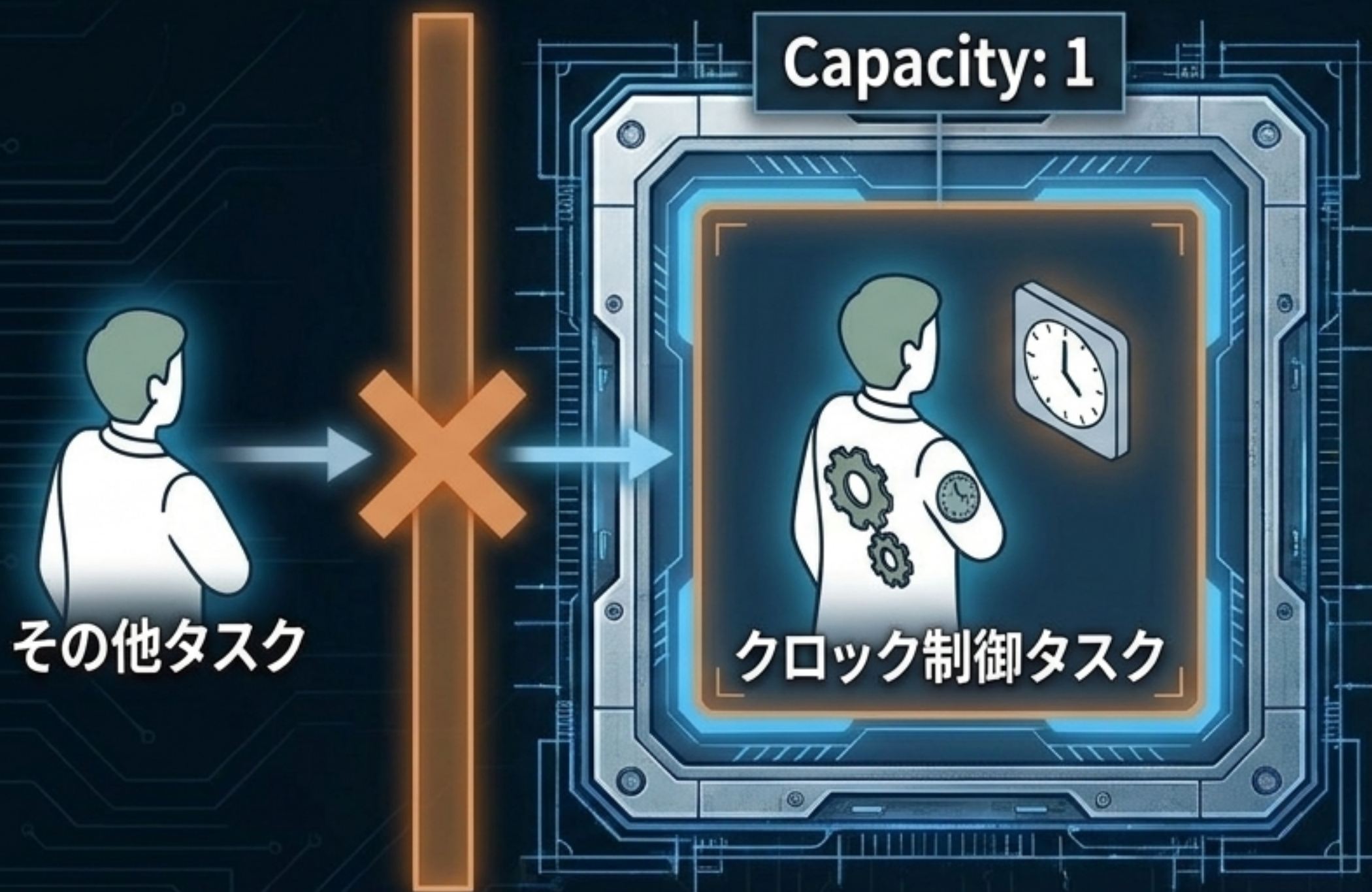
Analysis:

(1)で導き出した通り、このタスクの**優先度**はシステム内で**最も低い**。
それが現在「**CPUを占有している**」という事実が意味するシステム全体の状況を、**消去法**で割り出していく。



クロック制御タスク

Q3(2) Diagnostic 1: 「実行状態」の可能性



Diagnostic Rule Check:

CPU (MPU) が同時に処理できるタスクは「1つ」だけ。

Deduction:

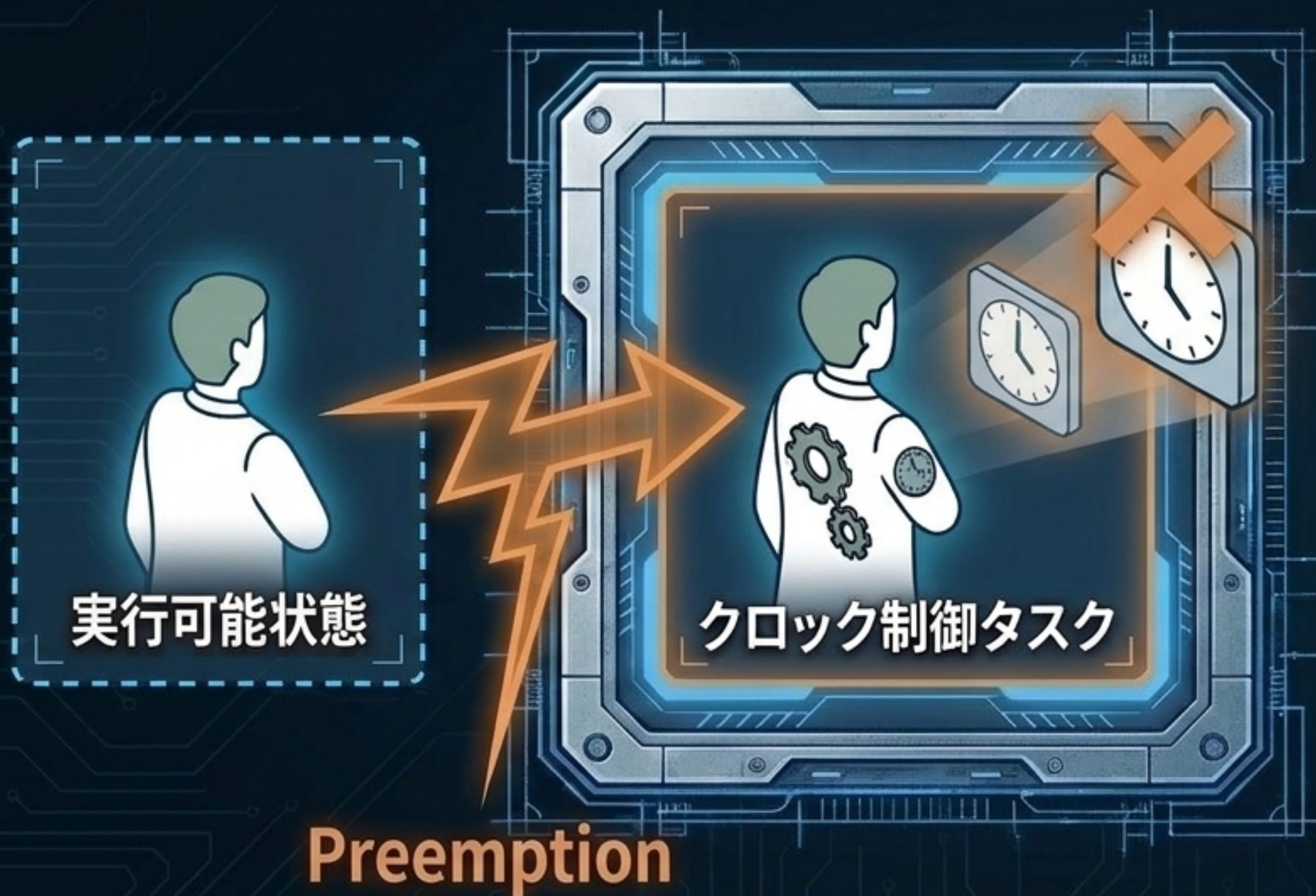
すでにクロック制御タスクが「実行状態」の枠を埋めている。

Result:

物理的制約により、ほかのタスクが「実行状態」に共存することは【あり得ない】。



Q3(2) Diagnostic 2: 「実行可能状態」の可能性



Diagnostic Rule Check:

(1)の結論により、ほかのタスクは全てクロック制御タスクより「優先度が高い」。

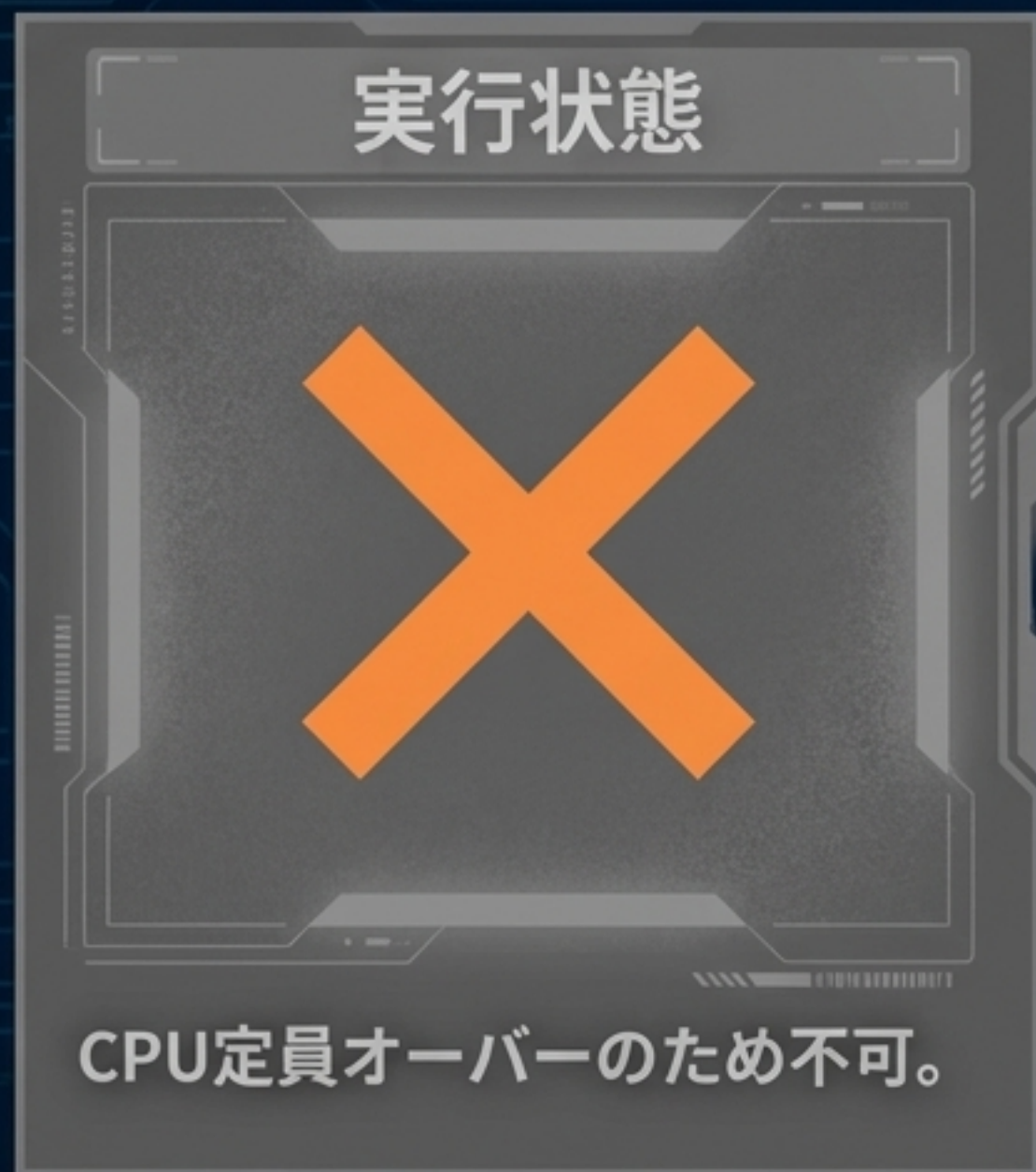
Deduction:

もし「実行可能状態」に他のタスクが存在、OSのルール(プリエンプション)が発動し、即座にCPUを奪い取るはずである。

Result:

クロック制御タスクが「実行され続けている」という事実こそが、実行可能状態のタスクが【1つも存在しない】証拠である。✗

Q3(2) Final Deduction: 消去法の終着点



他の全タスクは「メールの受信要求を出して、次の仕事に来るまで休眠している」状態である。残された最後の状態は？

Logic Completed: すべてのピースは揃った



- ✓ OSの絶対ルール「プリエンプション」と優先度の関係
- ✓ システムの本質「主役の仕事がない時に裏方が動く」
- ✓ 矛盾を突く「状態遷移の消去法」

これでシステム制御の「思考回路」は構築されました。
さあ、あなた自身の言葉で、解答用紙という設計図を完成させてください。