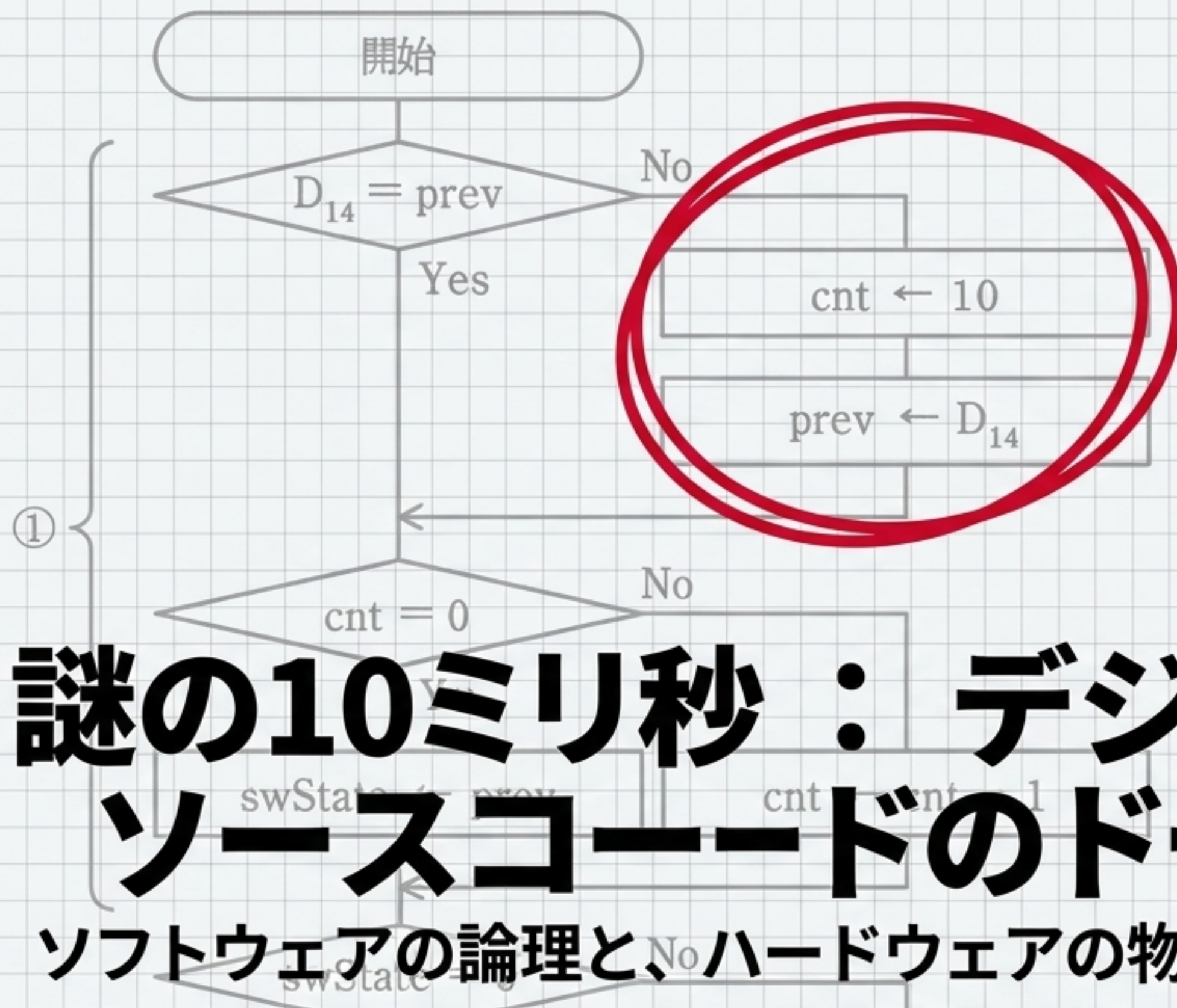




謎の10ミリ秒：デジタル時計の ソースコードのドーに潜む意図

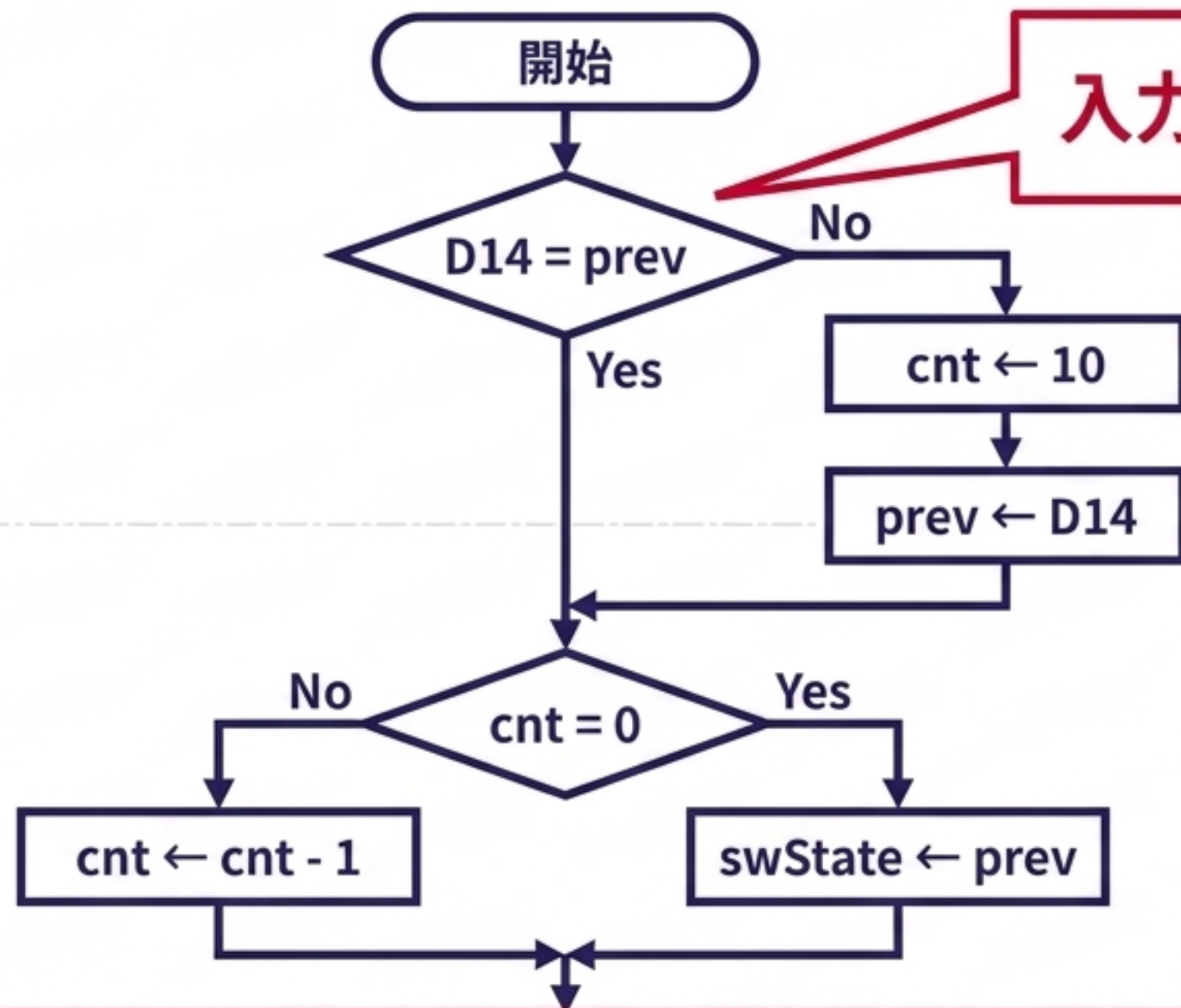
ソフトウェアの論理と、ハードウェアの物理が衝突する境界線を探る

平成23年度 春期 応用情報技術者試験 午後 問7 解析レポート

work ← a[i]

work ← b[i]

流れ図に隠された「不自然な待機」

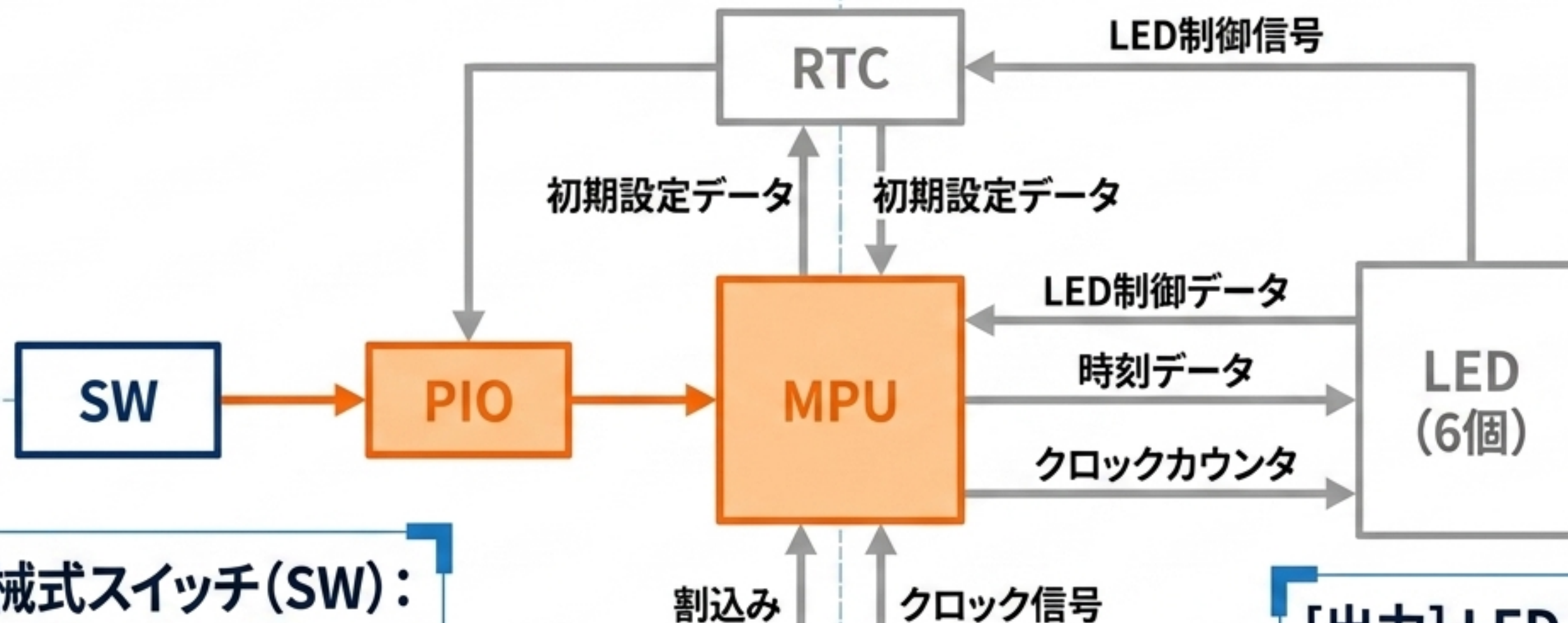


入力状態が変化したことを検知。

しかし、直ちに変更を受け入れない。
わざわざカウンタを「10」に設定し、
同じ状態が連続して確認されるまで
待機している。

なぜシステムは、目の前の入力を即座に「信用」しないのか？

システムの全体像：対象は「デジタル時計」

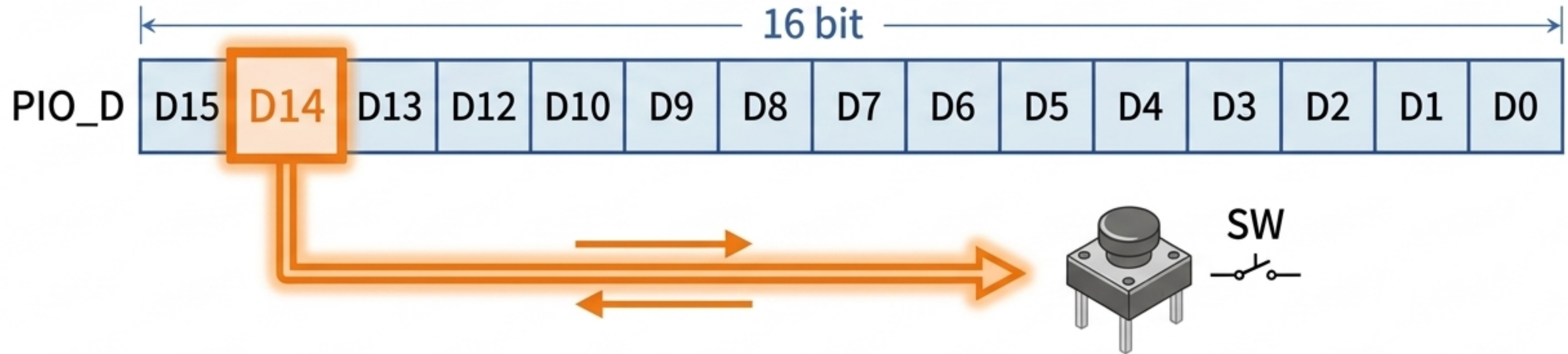


[入力] 機械式スイッチ(SW):
押下状態を物理的に切り替える。

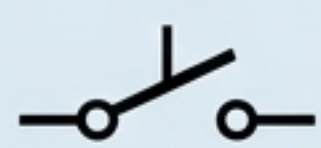
[処理] MPU:
システムの中枢。SWの状態に
応じて処理を分岐する。

[出力] LED(6個):
SWが押されている間は「年月日」を、離されている間は「時分秒」を表示する。

ゲートキーパー：PIOの「14番目の扉」



SW OFF: OFF



スイッチが離されている時、
D14 = 0 (時分秒を表示)

SW ON: ON



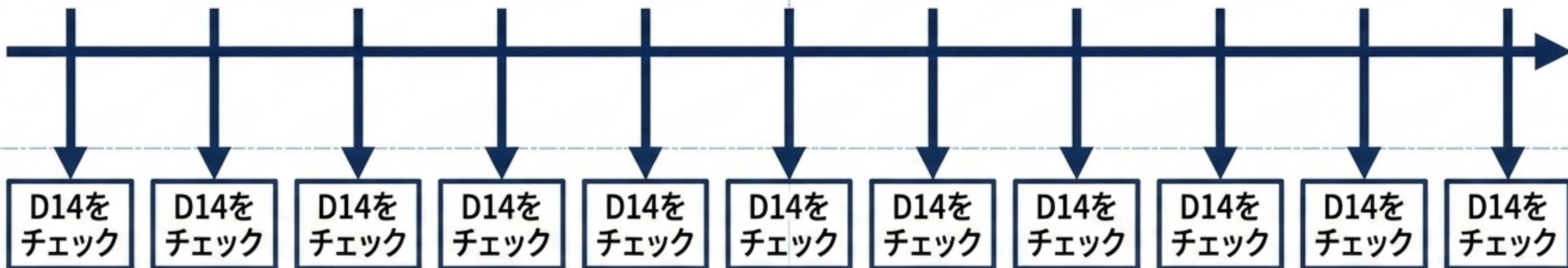
スイッチが押されている時、
D14 = 1 (年月日を表示)

MPUはスイッチを直接触ることはできない。
PIOのデータレジスタ (PIO_D) のD14を通じてのみ、世界を観測する。

容赦なき「1ミリ秒」の鼓動

Engineering grid

1ミリ秒 (1ms) 1ミリ秒 (1ms) 1ミリ秒 (1ms) 1ミリ秒 (1ms) 1ミリ秒 (1ms) 1ミリ秒 (1ms) 1ミリ秒 (1ms) 1ミリ秒 (1ms) 1ミリ秒 (1ms) 1ミリ秒 (1ms)



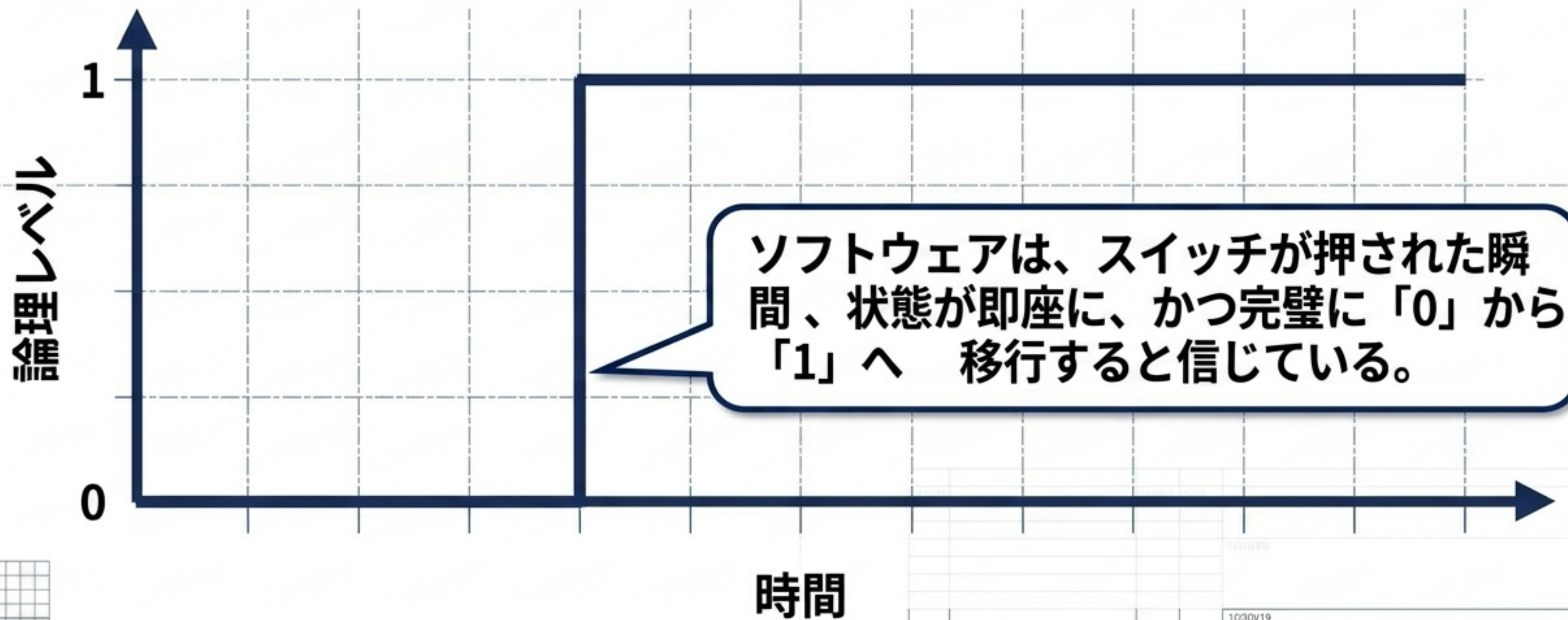
クロックカウンタは、正確に1ミリ秒ごとにMPUへ割込みを発生させる。

MPUはメイン処理を止め、毎秒1,000回のペースでD14 (SWの状態) を監視し続ける。

ソフトウェアにとっての時間は、極めて速く、そして厳格である。

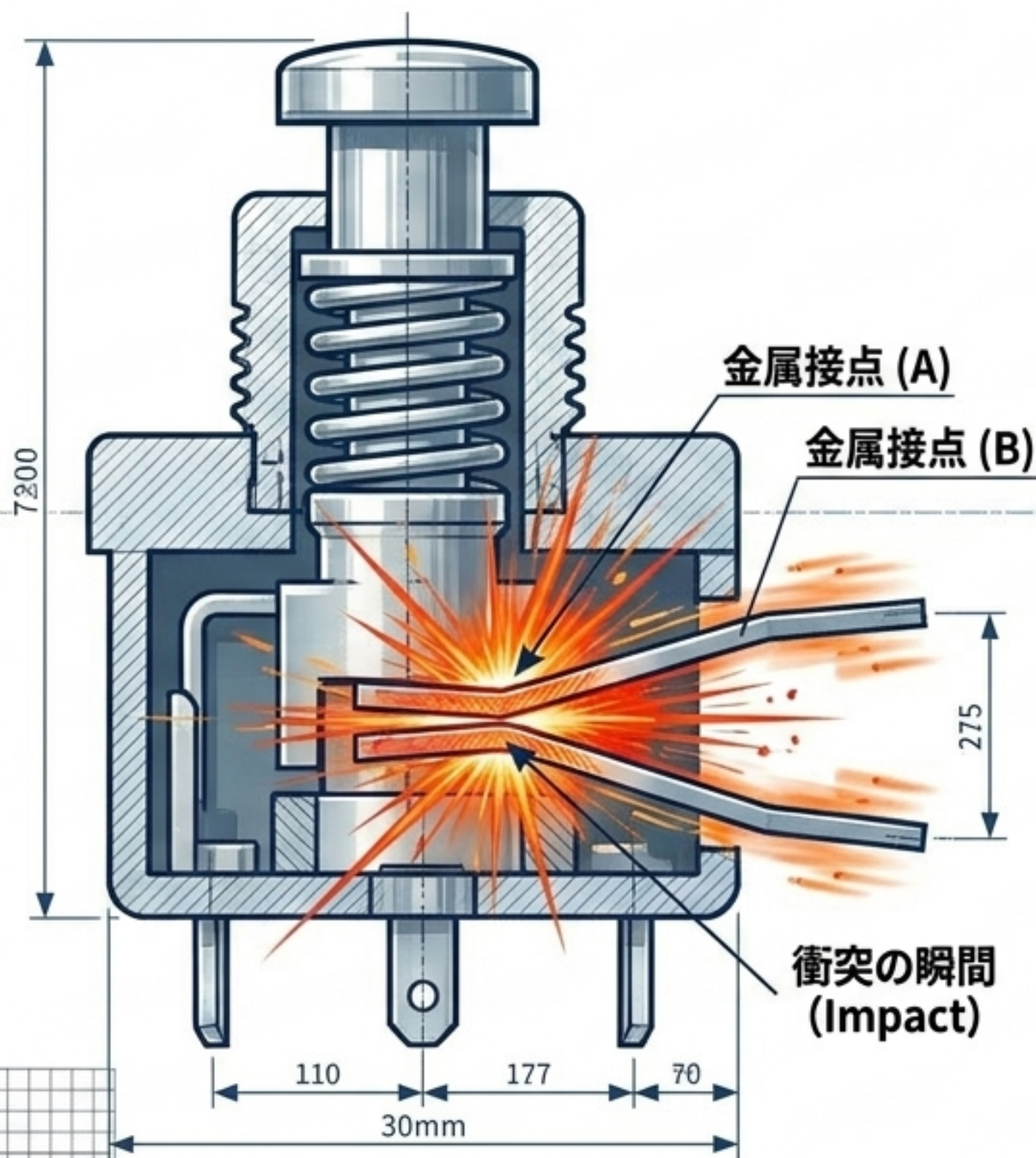
ソフトウェアが見る「理想の世界」

デジタルの世界では、物事は「0」か「1」のどちらかしかない。



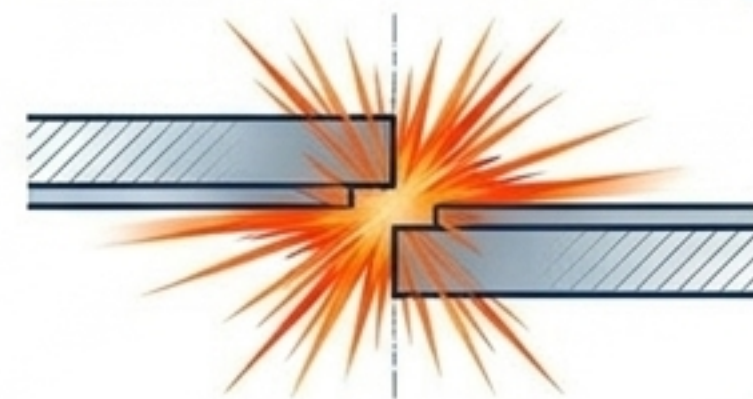
マイクロ秒単位の「物理的な現実」

Engineering grid



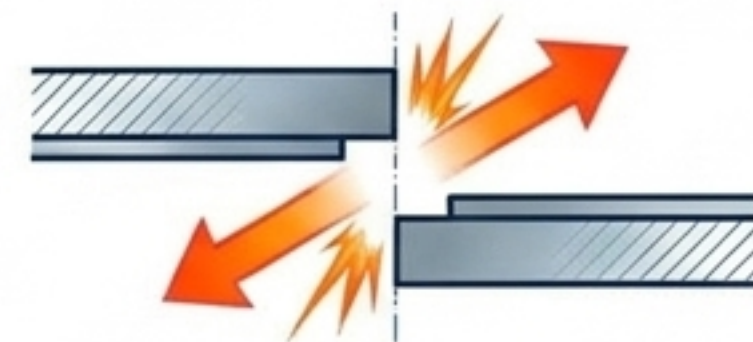
1. 衝突

スイッチを押すということは、内部で金属部品同士を物理的にぶつけることである。



2. 反発

固い金属同士が高速で衝突すると、微小なレベルで跳ね返り（バウンド）が生じる。

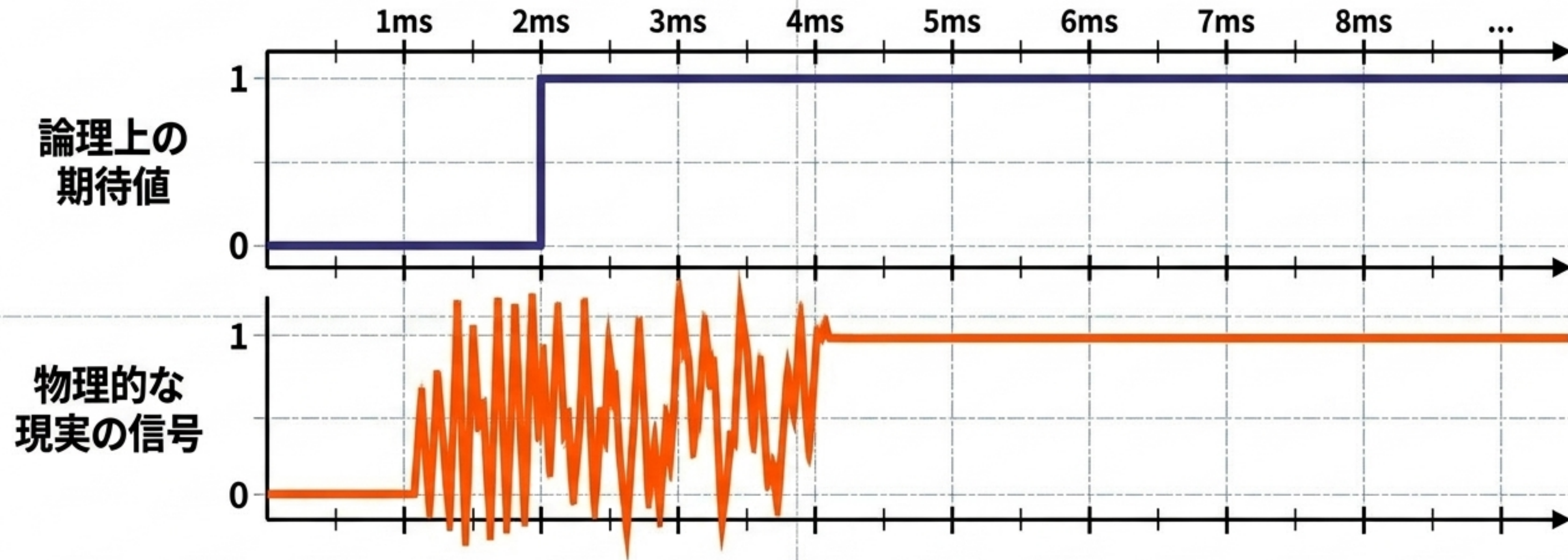


3. 微細な接触と離脱

金属が完全に密着するまでの数ミリ秒間、接点は「接触」と「離脱」を高速で繰り返してしまう。



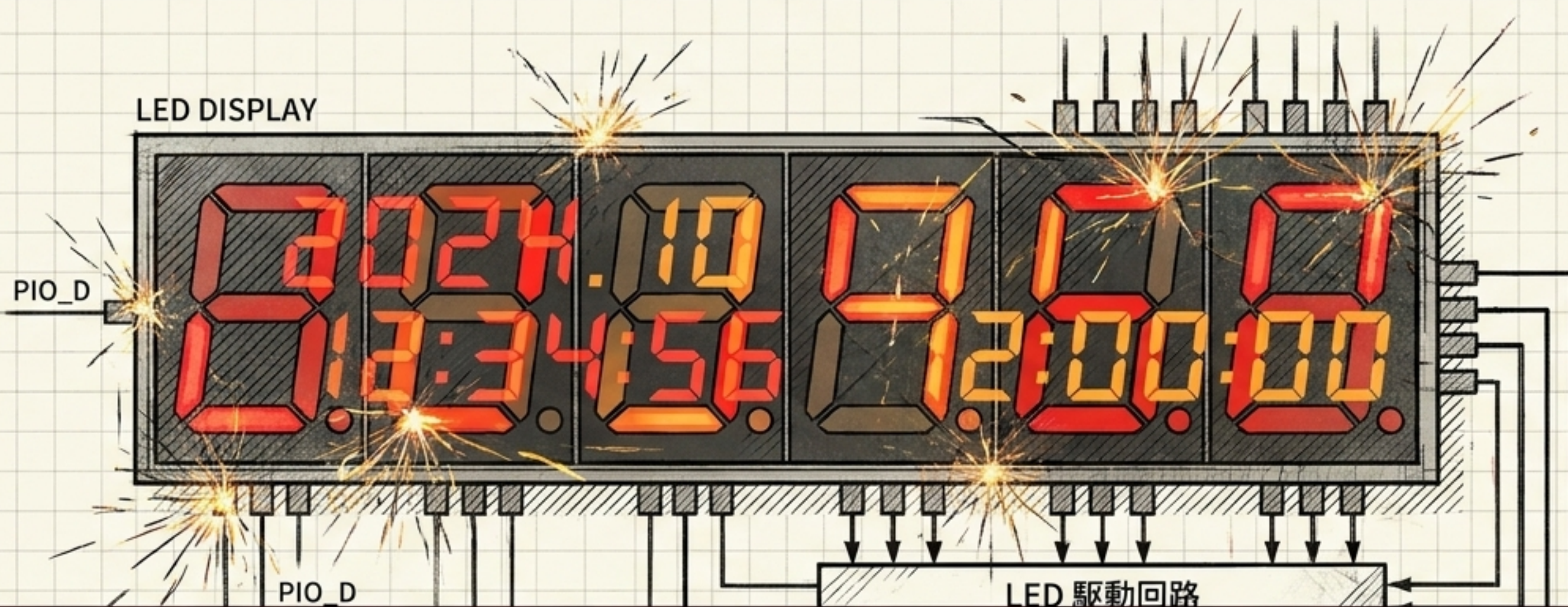
理想と現実の衝突



1ミリ秒ごとに観測を行うMPUが、この「物理的な波」をそのまま読み取るとどうなるか？

MPUの観測結果 = $[0, 1, 0, 1, 1, 0, 1, 1, 1, 1...]$

もし、この現実を「直視」してしまったら？

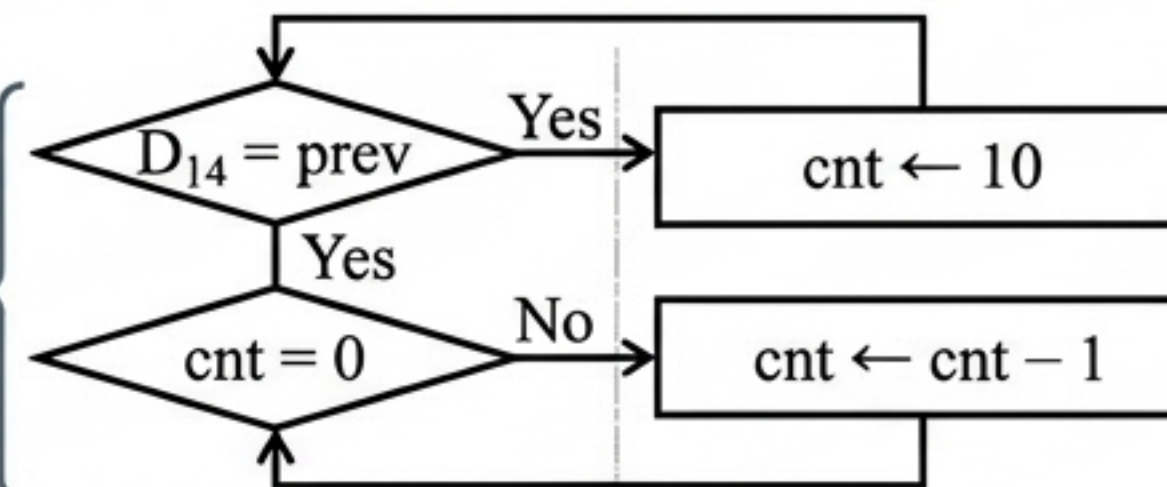


- 信号が [0] と [1] の間をバウンドしている最中に、システムが即座に反応した場合。
- 「時分秒」と「年月日」の表示が、1秒間に何百回も切り替わる。
- 結果として、ディスプレイは激しく点滅（フリッカー）し、使い物にならなくなる。

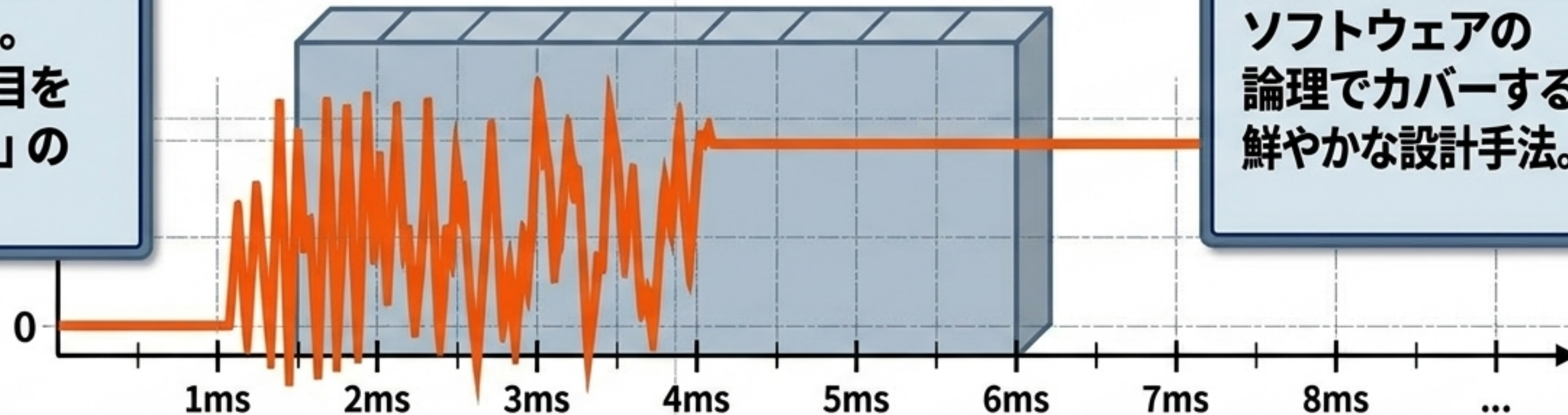
流れ図①の正体：物理現象に対するソフトウェアの盾

Engineering grid

このコードは、無駄に待っているのではない。意図的に「目を塞いでいる」のである。



ハードウェアの物理的な限界を、ソフトウェアの論理でカバーする鮮やかな設計手法。



10回 (10ミリ秒) 連続で同じ状態が確認されるまで待つことで、金属の物理的なバウンドが完全に収まるのを待機している。

設問3：図5中の①の処理で、SWの状態を直ちに反映しない理由を、30字以内で述べよ。

|

あなたは物理現象を知り、ソフトウェアの対応を知った。今、この現象に名前をつけ、理由をどのように30文字で要約するだろうか？