

Unterbrechungstransparenz

1. Welche grundsätzlichen Probleme bringt das Sperren von Interrupts zur Synchronisation mit sich? Welche Lösungswege zeichnen sich ab?
2. Was bedeuten die Begriffe „Prolog“ und „Epilog“, welche Unterschiede bestehen zwischen ihnen und welche verwandten Konzepte gibt es?
3. Welche Idee steht hinter dem Konzept der Schleusensynchronisation? Welche Aktionen laufen dabei ab beim Betreten und Verlassen eines kritischen Abschnitts?
4. Wie erfolgt die Abarbeitung zurückgestellter Epiloge? Weshalb implementieren die entsprechenden Anweisungsfolgen logisch gesehen einen kritischen Abschnitt und wie wird dieser geschützt? Unter welchen Randbedingungen sind die Anweisungsfolgen unkritisch?
5. Von wem und wie werden Epiloge ermöglicht und wann werden ermöglichte Epiloge abgearbeitet?
6. Was bedeutet die unterbrechungstransparente Epilogpropagation? Welche Effekte können ohne sie auftreten?
7. Weshalb ist die Festlegung des Reaktivierungszeitpunktes der Unterbrechungsebene₀ als besonders kritisch zu betrachten? Welche Lösungsmaßnahmen bieten sich hierzu an?
8. Was ist der Unterschied zwischen *level-triggered interrupts* und *edge-triggered interrupts*? Welche Art ist bei der unterbrechungstransparenten Epilogpropagation besonders problematisch? Was ist ein *spurious interrupt* und wie kann er entstehen?
9. Welches grundsätzliche Problem taucht bei der Rückkehr zum unterbrochenen Programm auf? Welche Lösungswege zeichnen sich hier ab?
10. Was bedeutet „*asynchronous system trap*“ (AST)? Wann wird ein AST im Zuge der Epilogpropagation ermöglicht und was wird dadurch bewirkt? Wie werden Epiloge bei der AST-Behandlung abgearbeitet?
11. Weshalb bildet die Epilogwarteschlange einen kritischen Datenbestand und wie wurde dieser synchronisiert? Warum brauchten dazu nur bestimmte (welche?) Unterbrechungsmuster berücksichtigt werden?