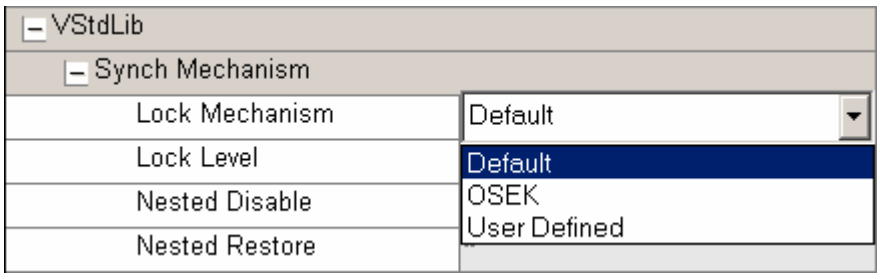
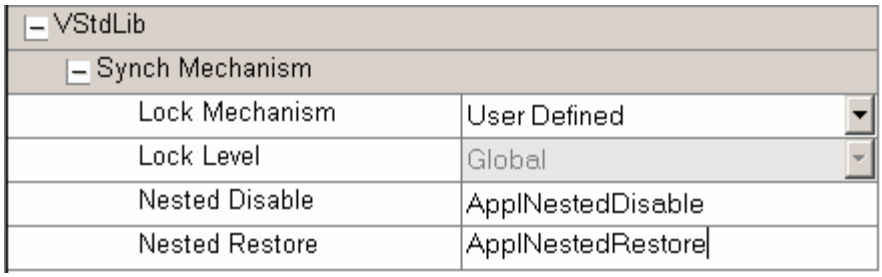
Application Interrupt Control with VStdLib

Version 1.0
2008-08-06
Application Note AN-ISC-2-1081

1.0 Overview

1.1 Introduction




VStdLib
Synch Mechanism
Lock Mechanism
User Defined
Lock Level
Global
Nested Disable
AppINestedDisable
Nested Restore
AppINestedRestore

2.0 Interrupt Control by Application

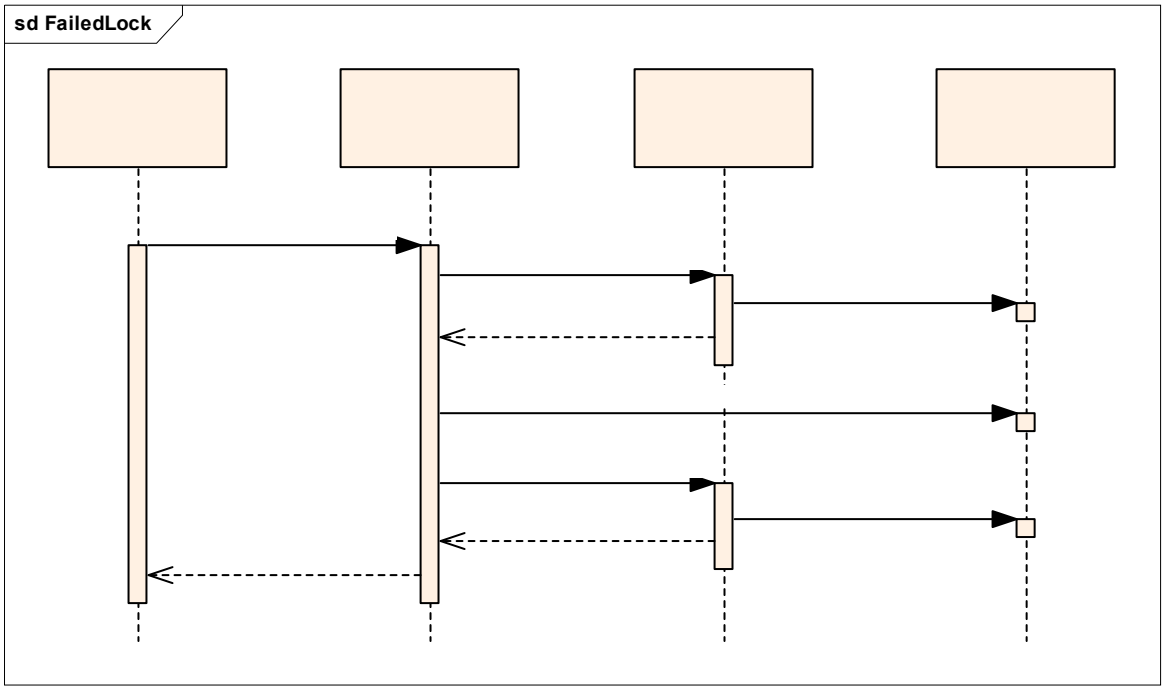
2.1 Constraints

2.1.1 Constraint 1: Nested Calls

2.1.2 Constraint 2: Recursive Calls when Disabling CAN Interrupts

2.1.3 Constraint 3: No Locking when Disabling CAN Interrupts

```
/* CAN Interrupt will be never locked in this example!!! */
void CanCanInterruptDisable(CAN_CHANNEL_CANTYPE_ONLY)
{
  ApplNestedDisable();
  Lock CAN interrupts
  ApplNestedRestore();
}

void ApplNestedDisable(void)
{
  Save current CAN interrupt state();
  Lock CAN Interrupts();
}

void ApplNestedRestore(void)
{
  Restore CAN interrupts to previous state();
}
```


sd FailedLock

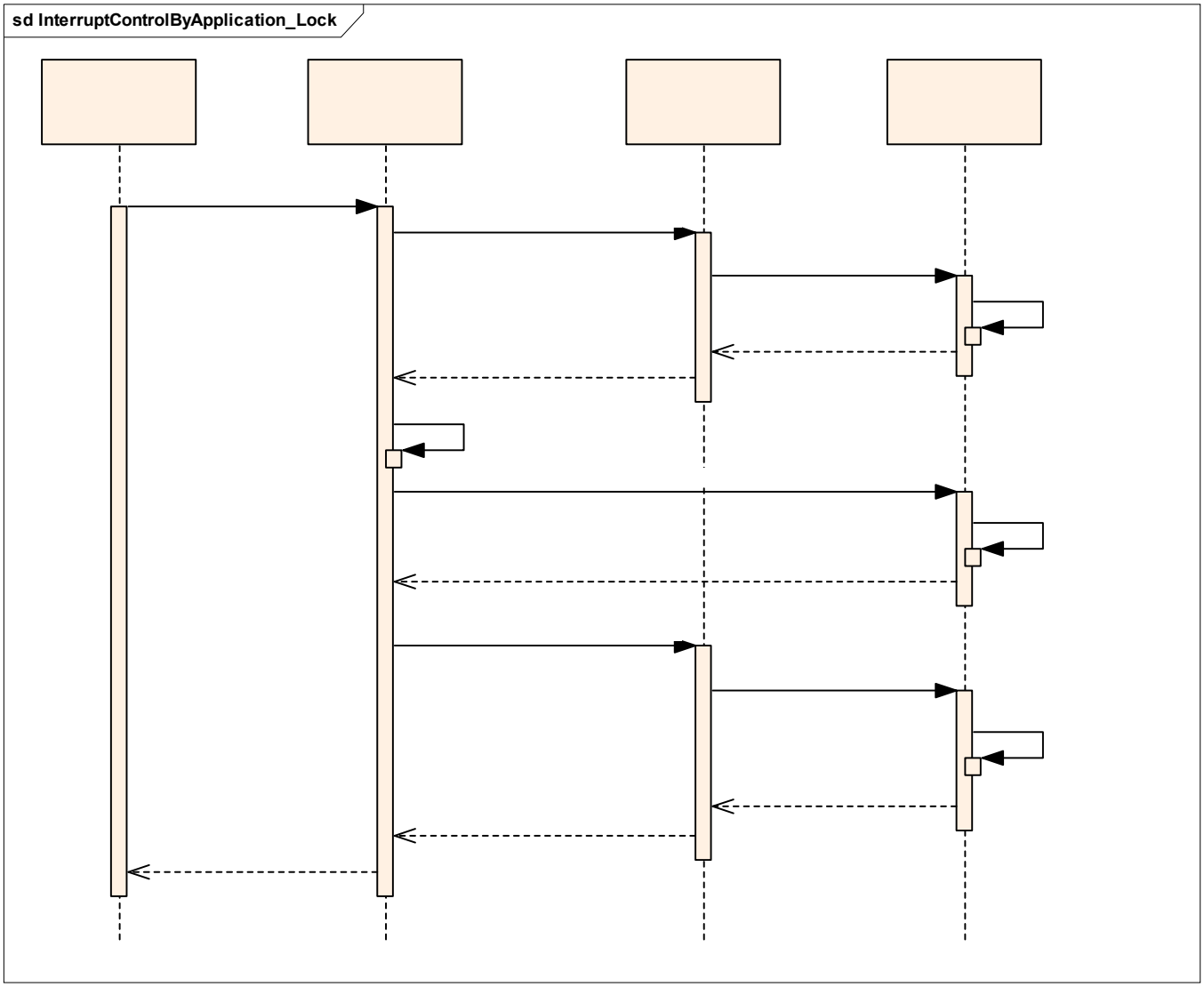
3.0 Solution

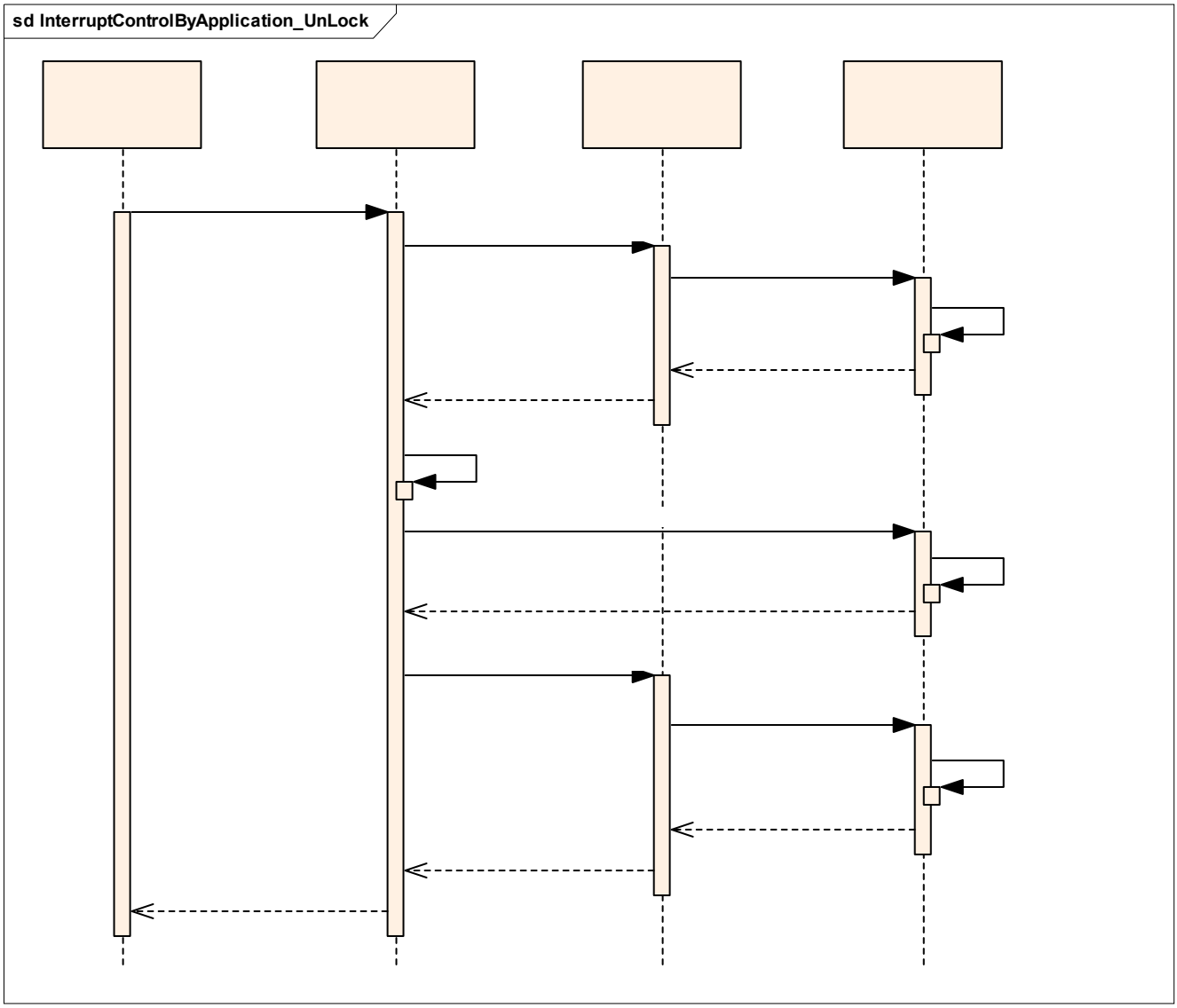
3.1.1 Nested Calls

```
/* Global variable as nesting counter */
vuint8 gApplNestingCounter;

/* Must be called before the Vector components are initialized! */
void SomeApplicationInitFunction(void)
{
  gApplNestingCounter = (vuint8)0;
}

void ApplNestedDisable(void)
{
  /* check counter - lock if counter is 0 */
  if((vuint8)0 == gApplNestingCounter)
  {
    /* Save current state and perform lock  */
    ApplicationSpecificSaveStateAndLock();
  }
  /* increment counter - do not disable if nested, because already done */
  gApplNestingCounter++;
}

void ApplNestedRestore(void)
{
  gApplNestingCounter--;
  if((vuint8)0 == gApplNestingCounter)
  {
```

```
    ApplicationSpecificRestoreToPreviousState();
  }
}
```

3.1.2 No Locking of Interrupts


sd InterruptControlByApplication_Lock


sd InterruptControlByApplication_UnLock

```
vuint8 gCanLockFlag;
vuint8 gApplNestingCounter;

void ApplicationInitFunction(void)
{
  /* initialize the flags */
  gCanLockFlag = (vuint8)0;
  gApplNestingCounter = (vuint8)0;
}

void ApplNestedDisable(void)
{
  if((vuint8)0 == gApplNestingCounter)
  {
    if((vuint8)0 == gCanLockFlag)
    {
      Save current CAN interrupt state();
      Lock CAN Interrupts();
    }
  }
  gApplNestingCounter++;
}

void ApplNestedRestore (void)
{
  gApplNestingCounter--;
  if((vuint8)0 == gApplNestingCounter)
  {
    if((vuint8)0 == gCanLockFlag)
    {
      Restore CAN interrupts to previous state();
    }
```

```
  }
}

void ApplCanAddCanInterruptDisable(CanChannelHandle channel)
{
  gCanLockFlag = (vuint8)1;
}

void ApplCanAddCanInterruptRestore(CanChannelHandle channel)
{
  gCanLockFlag = (vuint8)0;
}
```

4.0 Referenced Documents

Referenced Documents

5.0 Contacts

Vector Informatik GmbH	**Vector CANtech, Inc.**	**VecScan AB**
Vector France SAS	**Vector Japan Co. Ltd.**	**Vector Korea IT Inc.**