

2.7.1.5 Багатозначна логіка і IEEE 1164

У цій книзі ми обмежимося вбудованими системами, які мають вбудовану двійкову логіку. Тим не менш, може бути потрібно або необхідно використовувати більш ніж два значення для моделювання таких систем. Наприклад, наші системи можуть містити електричні сигнали різної сили, і це може бути необхідно для обчислення результуючої величини і логічного рівня в результаті з'єднання двох або більше джерел електричних сигналів. Надалі ми будемо розрізняти відмінність між рівнем і величиною сигналу. У той час як перше є абстракцією напруги сигналу, то останній є абстракцією опору джерела напруги. Ми будемо використовувати дискретну множину значень сигналів, що представляють рівень сигналу і величину. Використання дискретної величини дозволяє уникнути сильних проблем при розв'язуванні рівняння Кірхгофа і дозволяє нам уникнути аналогової моделі, яка використовується в електротехніці. Ми також будемо моделювати невідомі електричні сигнали за допомогою спеціальних значень сигналу.

На практиці, електронні системи використовують різні набори значень. Деякі системи дозволити тільки два, а інші дозволяють 9 або 46 значень. Основна мета розробки множини дискретних значень, це щоб уникнути проблем розв'язання рівнянь для електричного кола (наприклад, закони Кірхгофа) але ц до цих пір моделі існуючих систем мають достатню точність. Нижче ми представимо систематичний спосіб для нарощування множини значень які пов'язані один з одним. Ми будемо використовувати сили електричного сигналів в якості ключового параметра для розрізнення між різними множинами значень, це спосіб створення наборів значень, так звана теорія CSA, був представлений Hayes [Hayes, 1982]. CSA означає “connector, switch, attenuator”. Ці три елементи є ключовими елементами цієї теорії. Пізніше ми покажемо, як стандартна множина значень використовується в більшості випадків в VHDL.

1 сигнал (два логічних значення).

У простому випадку, ми почнемо з допомогою всього лише двох логічних значень - '0' і '1'. Ці два значення вважаються однієї і тієї ж величини. Це означає: якщо два проводи мають значення '0' і '1', то ми не будемо нічого знати про отриманий рівень сигналу. Одного сигналу може бути достаньо, якщо

два дроти, що мають значення '0' і '1' з'єднані і ніякі сигнали різної величин не відповідають певному вузлу електронної схеми.

2 рівня сигналу (три і чотири логічних значень)

У багатьох схемах, можуть бути випадки, в яких певний електричний сигнал не зумовлений ніяким виходом. Це може бути у випадку, коли певний провід не під'єднаний до землі, напруги живлення або будь-якого вузла схеми. Наприклад, системи можуть містити вихід з відкритим колектором (див. рис. 2.66, ліворуч). Для виходів з трьома станами (див. рис. 2.66, праворуч), дозволяючий сигнал '0' буде генерувати '0' на виходах затворів (позначається &) і обидва транзистори будуть непровідними. У результаті вихід буде від'єднаним. Таким чином, використовуючи відповідні входні сигнали, такі виходи можуть бути ефективно відключені від загального дроту.

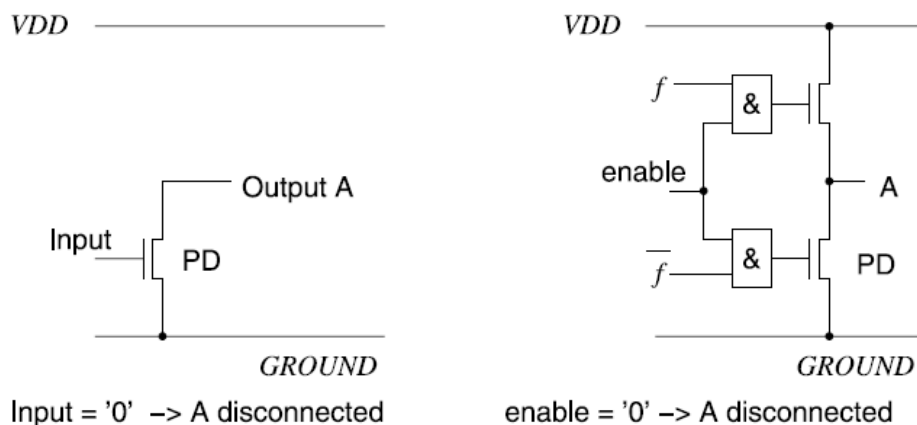


Рис. 2.66. Виходи можна ефективно відключити від загального дроту

Очевидно, що інтенсивність сигналу відключеного виходу має найменшу величину, що ми і можемо припустити. Зокрема, рівень сигналу Z менший, ніж '0' і '1'. Крім того, рівень сигналу такого виходу невідомий. Це поєднання сигналу і значення сигналу представлене логічним значенням, яке називається 'Z'. Якщо значення сигналу "Z" з'єднано з іншим сигналом, то інший сигнал завжди буде домінувати. Наприклад, якщо два трьохстанні виходи які з'єднані до однієї шини, і якщо один вихід сприяє значенню 'Z', то отримане значення на шині завжди буде значенням надане другим виходом (див. рис. 2.67). У VHDL кожен вихід пов'язаний з так званим signal driver. Обчислення значення результуючих драйверів до того ж сигналу називається дозволом і результуючі значення обчислюються з використанням функцій, які називаються функціями дозволу.

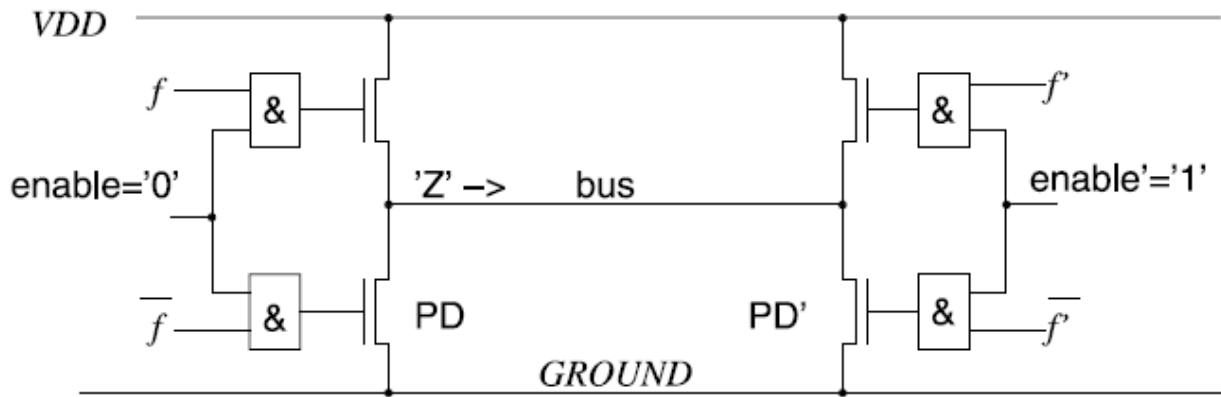


Рис. 2.67. Вихід правої домінуючої шини

У більшості випадків, тризначна логіка множин $\{0, 1, Z\}$ є розширеними четвертим значенням, яке називається 'X'. "X" являє собою невідомий рівень сигналу тієї ж сили, що '0' або '1'. Точніше, ми використовуємо 'X' для представлення невідомих значень сигналів, який може бути або '0' або '1' або деякі напруги, які не можуть бути ні '0' ні '1'. Резолюція, яка необхідна для того щоб підключити кілька драйверів, може бути обчислена дуже легко, якщо скористатися частковим порядком серед чотирьох значень сигналу '0', '1', 'Z' і 'X'. Частковий порядок зображено на діаграмі Хассе на рис. 2.68.

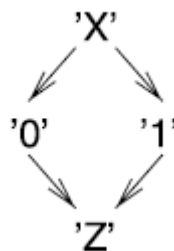


Рис. 2.68. Частковий порядок для множини сигналу '0', '1', 'Z' і 'X'

Ребрами у цьому зображенні відображають панування деякого значення сигналу. Краї визначають відносини $>$. Якщо $A > B$, то B домінує. '0' і '1' переважає 'Z'. 'X' переважає всі інші значення сигналу. На основі відношення $>$, визначимо відношення \geq . $A \geq B$ тоді і тільки тоді коли $A > B$ або $A = B$. Визначимо операцію *sup* двох сигналів, яка повертає верхню межу двох значень сигналу. Супремум з двох значень A і B є найслабшим значенням, для якого $C \geq A$ і $C \geq B$. Наприклад, $\text{sup}('Z', '0') = '0'$, $\text{sup}('Z', '1') = '1'$ і т.д. Цікавим спостереженням є те, що функції дозволу повинен обчислюють функції *sup* відповідно до наведених вище значень.

3 рівня сигналу (сім значень сигналу)

У багатьох схемах, двох сильних сигналів недостатньо. Часто виникає ситуація, яка вимагає більшого значення з використанням транзисторів виснаження (див. рис. 2.69).

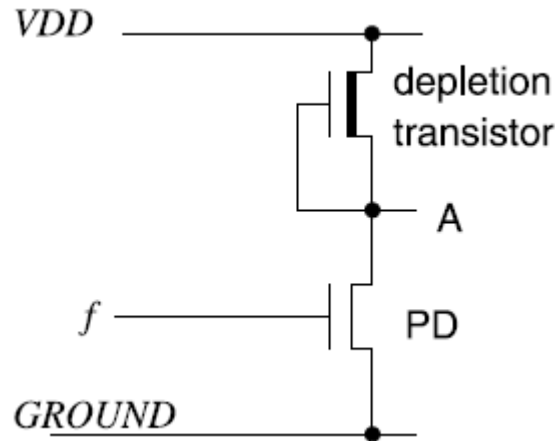


Рис. 2.69.

Ефект транзистори виснаження є близькою до того, як резистор забезпечує низьку провідність шляхом живлення VDD. Виснаження транзистора, а також "випадання транзистор" PD діють як драйвери для вузлів схеми і значення сигналу у вузлі можна обчислити за допомогою функції дозволу. Pulldown транзистори забезпечують драйвери, значення яких '0 'або' Z ', в залежності від вхідного сигналу PD. Транзистори виснаження забезпечують сигнал, значення якого є більш слабким, ніж '0 'і '1 '. Його рівень відповідає сигналу з рівнем '1 '. Ми представляємо значення надані транзистором виснаження 'Н', назовемо це «слабка логіка один». Подібно до цього, може бути слабкий логічний нуль, представлений 'L'. Результуюче значення можливого зв'язку між 'Н' і 'L' називається «визначеною слабкою логікою», яку позначають як "W ". У результаті, у нас є три сильних сигнали і сім логічних значень {0 ", '1 ', 'L ', 'Н ', 'W ', 'X ', 'Z '}. Дозвіл може бути здійснений на основі часткового порядку серед цих семи значень. Відповідний частковий порядок зображений на рис. 2.70.

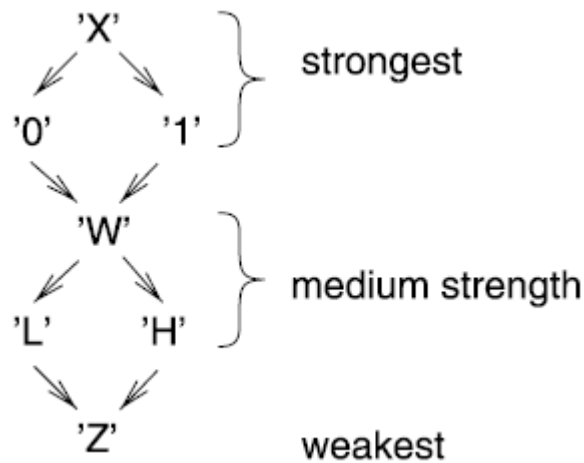


Рис. 2.70. Частковий порядок множини значень {0, 1, L, H, W, X, Z}

Цей порядок також визначає операцію *sup*, яка повертає найслабше значення серед двох сильних аргументів. Наприклад, $\text{sup}('H', '0') = '0'$, $\text{sup}('H', 'Z') = 'H'$, $\text{sup}('H', 'L') = 'W'$. '0' і 'L' мають ті ж рівні сигналу, але різної сили. Це також справедливо і для пар '1' і 'H'. Пристроями збільшення сили сигналу називають підсилювачами, пристрої зниження потужності сигналу називають атенуаторами.

Десять значень сигналу (4 величини сигналу)

У деяких випадках трьохрівневого сигналу є недостатньо. Наприклад, існує схема з використанням зарядів, накопичених на проводах. Такі дроти під'єднуються до рівнів '0' або '1' протягом декількох фаз роботи електронного ланцюга. Цей накопичений заряд може контролювати (високий опір) входи деяких транзисторів. Однак, якщо ці дроти підключені до найслабшого сигналу джерела (крім 'Z'), вони втрачають свій заряд і значення сигналу від цього джерела домінує.

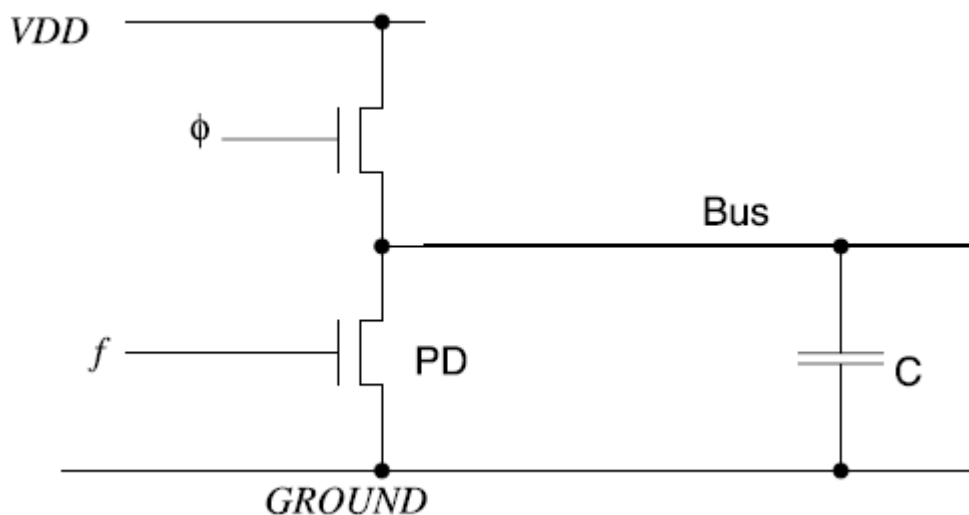


Рис. 2.71.

Автобус має високе ємнісне навантаження C . У той час як функції F як і раніше '0', покладемо ϕ рівним '1'. Для зарядки конденсатора C . Потім покладемо ϕ в '0'. Якщо реальне значення функції F стане відомо, і це, виявляється, '1', ми розряджуємо шину. Основна причина для використання попередньої зарядки є те, що зарядка шини з використанням виведення, такі як показано на рис. 2.69 є повільним процесом, тому що опір виснажувальних транзисторів є великим. Розрядка за допомогою регулярних pulldown транзисторів PD пришвидшує процес. Для моделювання таких випадків, ми повинні значення сигналу, які слабші, ніж 'H' і 'L', але сильніші, ніж 'Z'. Ми назвемо такі цінності "дуже слабкі значення сигналу" і позначимо їх 'H' і 'L'. Відповідне дуже слабе невідоме значення позначається 'W'. В результаті отримуємо десять значень сигналу {0, '1', 'L', 'H', '1', 'H', 'X', 'W', 'W', 'Z'}. Можна знову визначити частковий порядок серед цих значень (див. рис. 2.72).

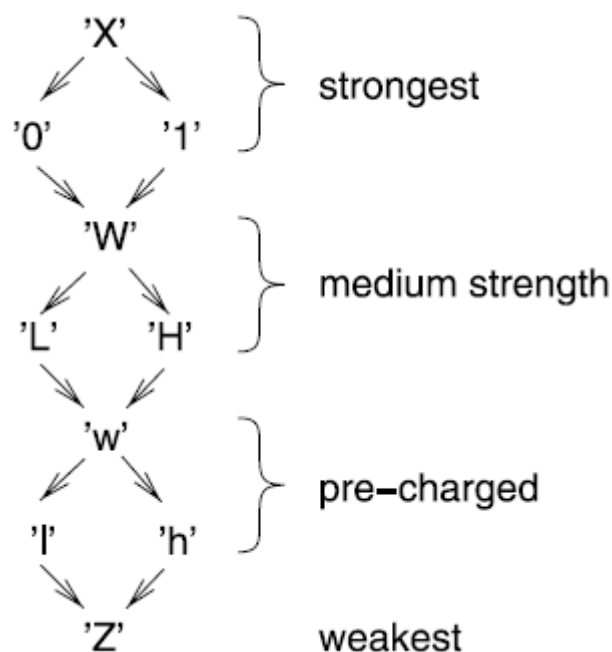


Рис. 2.72. Частковий порядок для множини {0, '1', 'L', 'H', '1', 'H', 'X', 'W', 'W', 'Z'}

IEEE 1164

У VHDL немає визначеної кількості значень сигналу, за винятком деяких базова підтримка двозначної логіки. Для спрощення обміну VHDL моделі, стандартна множина була визначена і стандартизована IEEE. Цей стандарт називається IEEE 1164 і використовується в багатьох моделях системи.

IEEE 1164 має дев'ять значень: {0, '1', 'L', 'H', 'X', 'W', 'Z', 'U', '-'}. Перші сім значення відповідають сімом значень сигналів описаних вище. 'U' позначає неініціалізовані значення.

'-' позначає вхідний сигнал. У VHDL вибір виразу є дуже зручним засобом для цього. Оператору SELECT відповідає вимикач і вираз у випадку пошуку в інших мовах і його значення відрізняється від оператора вибору в ADA (Див. стор 104).

Приклад: Припустимо, що ми хотіли б представляти булеву функцію:

$$f(a,b,c) = \overline{a}b + bc$$

Крім того, припустимо, що F повинна бути визначена для випадку A = B = C = 0. Завдання цієї функції будуть розподілені наступним чином:

```
f <= select a & b & c -- & denotes concatenation
      '1' when "10-" -- corresponds to first term
      '1' when "-11" -- corresponds to second term
      'X' when "000"
```

Таким чином, наведені вище функції можуть бути легко переведені на VHDL. На жаль, вибраний вираз означає щось зовсім інше. VHDL інструменти оцінюють такі вирази, як один описаних вище, вони перевіряють, якщо вибраний вираз (& B & C в попередньому випадку) рівний вибраним умовам.. Зокрема, вони перевіряють, якщо, наприклад, a & b& c рівне "10 -". У цьому контексті, '-' поводить ся як будь-яке інше значення: система VHD перевіряє коректність c має значення з "-". Оскільки "-" ніколи не призначені для будь-яких змінних, ці тести ніколи не будуть бути правдивими. Таким чином, '-' має обмежену користь. Хороша властивість загальної дискусії на сторінках 90 до 95 виглядає наступним чином: це дозволяє нам негайно зробити висновки про потужність моделювання IEEE 1164. IEEE стандарт заснований на 7-значних множинах значень, яке описано на стор 93 і, отже, здатна моделювання схеми, що містять виснажуючі транзистори.

2.7.2 System C

У зв'язку з тенденцією реалізації все більше і більше функціональності в програмному забезпеченні, більше число вбудованих систем включає суміш апаратних засобів і програмного забезпечення. Наприклад, вбудовані системи реалізують стандарти, такі як MPEG 1/2/4 або декодери для стандартів мобільних телефонів, таких як GSM або UMTS. Стандарти часто доступні у вигляді "еталоннів реалізації", що складається з C програми, що не оптимізовані для швидкості, але забезпечують необхідну функціональність.

Крім того, імітуючи апаратне і програмне забезпечення вимагає взаємодії програмного і апаратного забезпечення. Таким чином, була необхідність пошуку методів відтворення апаратних структур в галузі програмного забезпечення.

SystemC включає в себе поняття процесів, виконуваних одночасно. Семантика моделювання аналогічна VHDL, в тому числі наявність дельта циклів. Виконання цих процесів здійснюється через чутливі спис і заставляє чекати примітивів. Поняття чутливості спису включає в себе динамічні списи чутливості. Раніше SystemC 1.0 використовувала числа з плаваючою комою. В даний час стандартом є цілочисельна модель часу яка підтримує фізичні одиниці, такі як пікосекунди, наносекунди, мікросекунд і т.д.

Типи даних SystemC включають в себе всі поширені апаратні типи: чотиризначної логіки ('0 ', '1 ', 'X 'і 'Z ') і бітвекторів різної довжини. Написання програм для цифрової обробки сигналів спрощується завдяки наявності типів даних з фіксованою точкою.

В цілому не гарантується визначеної поведінки, допоки не буде використано конкретний стиль моделювання. Використовуючи командний рядок, симулятор може бути використаний для запуску процесів в різних порядках. Таким чином, користувач може перевірити залежать результати моделювання від послідовності, в якій процеси були виконані.

Повторне використання апаратних компонентів у різних контекстах спрощується за рахунок розділення обчислень і комунікацій. SystemC надає канали, порти, 98 вбудованих систем і абстрактні інтерфейси компонент зв'язку.

Визначення: "**Transaction-level modeling (TLM)** є підходом високого рівня до моделювання цифрових систем, в яких деталі зв'язку між модулями відокремлені від деталей реалізації функціональних блоків або архітектури зв'язку. Механізми комунікації таких, як автобуси або FIFOs моделюються як канали і представлені в модулях, які використовують інтерфейс SystemC.. На рівні транзакцій, акцент робиться більше на функціональності передачі даних - те, що дані передаються і з якого місця - і менше на їх фактичній реалізації, тобто про протокол, використовуваний для передачі даних. Такий підхід полегшує експериментувати для системного дизайнера, наприклад, з різною архітектурою шини без необхідності перебудови моделей, які взаємодіють з будь-якою з шин, якщо ці моделі взаємодіють з шиною через загальний інтерфейс.

2.7.3 Verilog та System Verilog

Verilog це інша мова опису апаратних засобів. Спочатку вона була патентованою мовою, але пізніше стандартизована як IEEE стандарт 1364, з версіями, які називаються стандарт IEEE 1364-1995 (Verilog версія 1.0) та стандарт IEEE 1364-2001 (Verilog 2.0). Деякі особливості Verilog дуже схожі на VHDL. Так само, як в VHDL, конструкції описані у вигляді набору підключених об'єктів дизайну, і дизайн об'єктів може описуватися поведінково. Крім того, процеси використовуються для моделювання паралелізму апаратних компонентів. Як і в VHDL, бітвектори та одиниці часу підтримуються. Є, однак, деякі області, в яких Verilog менш гнучкий і більше фокусується на зручні вбудовані функції. Наприклад, стандарт Verilog не включає в себе гнучкі механізми для визначення перерахованих типів, таких як ті, що визначені в IEEE 1164 стандарті, підтримка чотирьох-значної логіки побудована на мові Verilog, і стандартний IEEE 1364 також надає кілька значної логіки з 8 різними рівнями сигналу. Багатозначна логіка, більш тісно інтегрована в Verilog ніж у VHDL. Система Verilog логікитакож надає більше можливостей для транзисторного рівня опису. Тим не менш, VHDL є більш гнучким. Так, наприклад, VHDL дозволяє, щоб апаратні сутності бути оброблені в петлях. Це може бути використано, щоб створити структурні описи, наприклад N-бітного суматора без вказівки N суматорів та їх взаємозв'язків вручну.

Verilog має аналогічну VHDL кількість користувачів. У той час як VHDL є популярнішим в Європі, Verilog є популярнішим в США. Verilog версій 3.0 і 3.1, також відомий як SystemVerilog. Вони включають в себе численні розширення Verilog 2.0. Ці розширення включають в себе [Accellera Inc., 2003], [Sutherland, 2003]: додаткові елементи мови для моделювання поведінки, С типи даних, такі як `int` та визначення типів об'єктів, таких як `typedef` і `struct`, визначення інтерфейсів апаратних компонентів як окремих сутностей, стандартний механізм виклику `C / C++` функцій і, в деякій мірі, вбудованих Verilog функції з `C`, значно вдосконалені функції для опису середовища (так званий випробувальний стенд) для апаратної схеми під дизайн (так званий КЕД), а також для використання випробувального стенда для перевірки КЕД шляхом моделювання, класи відомі з об'єктно-орієнтованого програмування для використання на випробувальному стенді, динамічне створення процесу, стандартизований механізм взаємодії та синхронізації, в тому числі семафорів, автоматичне виділення і звільнення пам'яті, властивості мови, які забезпечують стандартний інтерфейс для формальної перевірки (Див. ст. 203). У зв'язку з можливістю взаємодії з `C` і `C++`, взаємодія з SystemC моделей є також можливою. Поліпшені засоби для моделювання, а також для формального аналізу на основі перевірки дизайну і можливим інтерфейсом з SystemC потенційно справляють дуже гарне враження. Останнім часом і Verilog та SystemVerilog були об'єднані в один стандарт, IEEE 1800-2009 [IEEE, 2009].