**4. Розробка функціональної схеми**

За технічним завданням необхідно було розробити функціональну схему інтерфейсу зовнішнього пристрою, що містить в собі розподілені арбітри пріоритетних переривань і прямого доступу до пам'яті.

Завдання було реалізоване на схемі функціональної електричною ІЛАЦ.463626.005 Е2. Розглянемо докладніше функціонування схеми.

Регістр 1 є регістром адреси. Він необхідний для того, що в системі використовується суміщена шина адреси і даних, і необхідно запам'ятати адресу, щоб потім знати, як обробляти що поступають услід дані. Регістр 1 є 7-міразрядним, оскільки адреса для зовнішньої пам'яті даних є 7-миразрядним числом.

Старші 7 біт адреси далі подаються на селектор адреси (елемент №13), де відбувається аналіз, чи до даного пристрою йде звернення. Входи у цього елементу І можуть бути як прямими, так і інверсними, обов'язково різними для кожного зовнішнього пристрою, оскільки задають адресу в адресному просторі пам'яті даних, по якому можна отримати інформацію саме з цього конкретного зовнішнього пристрою. Вихід з селектора адреси далі використовується для блокування/дозволу роботи решти елементів схеми.

Молодший біт адреси з Регістра 1 служить для визначення, до якого саме регістру ми звертатимемося - регістру даних або регістру станів. Регістру даних відповідає «0» в цьому біті, а регістру стану - «1».

На інвертори 14 і 15 подаються відповідно сигнали IN і OUT. Елементи 20-23 визначають операцію і регістр, з яким вона проводитиметься з урахуванням перевірки, чи вибраний поточний зовнішній пристрій.

Елемент 20 відповідає за сигнал читання з регістра даних зовнішнього пристрою. Він спрацьовує, якщо пристрій вибраний (аналізуємо сигнал з селектора адреси), якщо адреса відповідає регістру даних (останній біт 0) і поданий сигнал читання (IN). Його вихід підключається до входу OE (Output enabled) регістра даних (елемент №3).

Елемент 21 відповідає за сигнал читання з регістра станів зовнішнього пристрою. Він спрацьовує, якщо пристрій вибраний (аналізуємо сигнал з селектора адреси), якщо адреса відповідає регістру станів (останній біт 1) і поданий сигнал читання (IN). Його вихід підключається до входу OE (Output enabled) регістра станів (елемент №4).

Елемент 22 відповідає за сигнал запису в регістр даних зовнішнього пристрою. Він спрацьовує, якщо пристрій вибраний (аналізуємо сигнал з селектора адреси), якщо адреса відповідає регістру даних (останній біт 0) і поданий сигнал запису (OUT). Його вихід підключається до входу W регістра даних (елемент №3).

Елемент 23 відповідає за сигнал запису в регістр стану зовнішнього пристрою. Він спрацьовує, якщо пристрій вибраний (аналізуємо сигнал з селектора адреси), якщо адреса відповідає регістру стану (останній біт 1) і поданий сигнал запису (OUT). Його вихід підключається до входу W регістра стану (елемент №4).

Затримки 44 і 45, а також елементи І-НЕ №25-26 захищають сигнали запису від просічень - короткочасних стрибків сигналів. У такому разі в регістр відповідно даних або стану може записатися випадкова інформація, що може привести до некоректної роботи системи або окремого зовнішнього пристрою.

Елемент №37 (ІЛІ-НЕ) визначає, чи проходить зараз операція читання або запису над регістрами. Цей сигнал важливий для схем обробки переривань і прямого доступу до пам'яті. Елемент затримки 43 захищає цей сигнал від просічень.

Елементи 24 і 38 відстежують ситуацію, коли вибраний пристрій і його регістр станів, але при цьому не відбувається операцій читання/запису, і пристрій знаходиться в стані дозволу переривання. Сигнал з елементу 38 йде на вхід вибору напряму передачі шинного формувача 12. Цією частиною схеми визначається ситуація, коли на шину адреса/данних треба подавати інформацію з вектора переривання. У решті всіх ситуацій шинний формувач відкритий для запису в пристрій.

Регістр 3 - регістр даних зовнішнього пристрою. Він доступний по читанню/запису як елемент зовнішньої пам'яті даних за певною адресою. Його вихід йде на шину адреса/данних. Запис в цей регістр відбувається через шинний формувач по сигналу з елементу 25.

Регістр 4 - регістр стану зовнішнього пристрою. Він доступний тільки по читанню як елемент зовнішньої пам'яті даних за певною адресою. Його вихід йде на шину адреса/данних. Запис в цей регістр походить з самого пристрою по сигналу з елементу 26 (тобто само пристрій визначає зміна в своєму стані). Битий 0 визначає готовність пристрою, битий 1 - вимога переривання, битий 2 - вимога прямого доступу до пам'яті.

Розглянемо, як працює розподілений арбітр пріоритетних переривань. Елемент 29 аналізує, чи одночасний сигнал готовності і вимоги переривання. Трігер 10 перемкнеться в «1». Елемент №20 перевіряє ще раз, що сигнал на вході синхросигналу і виході трігера 10 однаковий і рівний «1». Далі цей сигнал йде на шину вимоги переривання. Трігер 10 скидається в «0» під час вступу нульового сигналу на елемент 16 з входу 94. Елемент 27 відповідає за передачу сигналу наступному по пріоритету зовнішньому пристрою. Цей сигнал буде рівний «0», тобто заборона переривання, у випадку якщо, один з попередніх пристроїв або те, що розглядається вимагає переривання. Це буде формування дейзи-цепочки з елементів по пріоритетах. На елемент 17 приходить підтвердження переривання від мікроконтролера. Далі сигнал йде на трігер 11, який перемикається в «1» за наявності підтвердження переривання і вимоги переривання (на трігері 10). Вихід елементу 32 рівний «1» у випадку, якщо пристрій отримало підтвердження переривання і по пріоритету з дейзи-цепочки переривання покладене саме йому. Далі з елементу 32 сигнал йде на фільтр 40, далі - на елемент 31, який видає остаточний сигнал переривання. Елемент 40 видасть сигнал переривання на контроллер у випадку, якщо пристрій тредуєт переривання, отримано підтвердження переривання, немає пріоритетніших пристроїв, що вимагають переривання, і в даний момент з регістрами пристрою не відбуваються операції читання/запису.

По сигналу переривання (вихід елементу 32) на шину адреса-данних подається сигнал з регитра вектора переривання (елемент №2). Номер зовнішнього пристрою набирається на його входах за допомогою спеціальних перемикачів.

Дейзи-цепочка контроллера прямого доступу до пам'яті працює аналогічно. Трігер 9 визначає наявність вимоги прямого доступу до пам'яті, трігер 8 визначає підтверджену вимогу прямого доступу. З виходу елементу 36 йде сигнал на схему управління пам'яттю, з виходу елементу 41 - сигнал вимоги прямого доступу до пам'яті на мікропроцесор. Сигнал з виходу 28 подається на модуль арбітра прямого доступу до пам'яті наступного по пріоритету пристрою.

Регістр 5 є регістром початкової адреси в пам'яті, по якій необхідно пересилати масив байт. На його вхід подається інформація від блоку управління пристрою введення/виводу, вихід йде на схему управління пам'яті. Регістр 6 визначає номер зовнішнього пристрою, в який необхідно виконати пересилку. Лічильник 7 визначає кількість слів, яку необхідно переслати, використовуючи механізм прямого доступу до пам'яті.

**5. Розробка принципової схеми**

Принципова схема повністю відповідає функціональною. Відповідно призначення елементів на ній слід дивитися в описі відповідних елементів функціональної схеми (див. п.4.).

**Виводи**

У даній курсовій роботі приведений весь процес розробки мікропроцесорної системи - від складання структурної схеми за технічним завданням до розробки принципової схеми конкретних вузлів і написання програми для обчислення необхідної функції.

При розробці структурної схеми була показана можливість істотно розширити можливості мікропроцесора шляхом підключення до нього додаткових модулів.

При розробці програмного продукту була продемонстрована необхідність структурувати складний програмний продукт для зручності його розробки, читабельності коди і полегшення процесу виправлення помилок. З цією метою розробка програми була зведена до 3 етапів - розробки загальної концепції розміщення операндів і обміну інформацією між програмними модулями, розробки набору вспомагательних процедур нижнього рівня і написанні необхідних процедур високого уровни з використанням процедур ніжнего.

Також було розглянуто місце використовуваного контроллера серед сучасних рішень.