

10. Построение схем с иерархией.

С этой главы мы начинаем большую тему, посвящённую построению иерархических схем в OrCAD Capture.

10.1. Flat Design.

Итак, один из способов организации проекта в OrCAD Capture – «двумерный» или «сплошной», который носит название **Flat Design**. В этом случае, дизайн состоит из одной папки Schematic. Схема может быть расположена на нескольких листах.

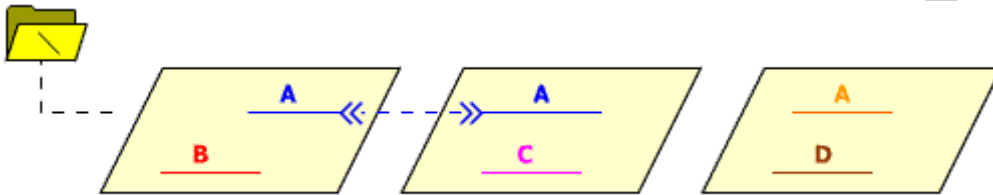


рис. 10-1

На **рис. 10-1** показана структура дизайна, состоящего из трёх листов. На каждом листе условно изображены по две цепи. Цепи **A**, расположенные на первом и втором листах соединяются через Off-Page Connector и, таким образом, являются одним целым. Цепь **A**, которая находится на третьем листе, Off-Page Connector-а не имеет, и, значит, будет оторвана от первых двух.

Рассмотрим на примере. OrCAD Capture дизайн (**рис. 10-2**):

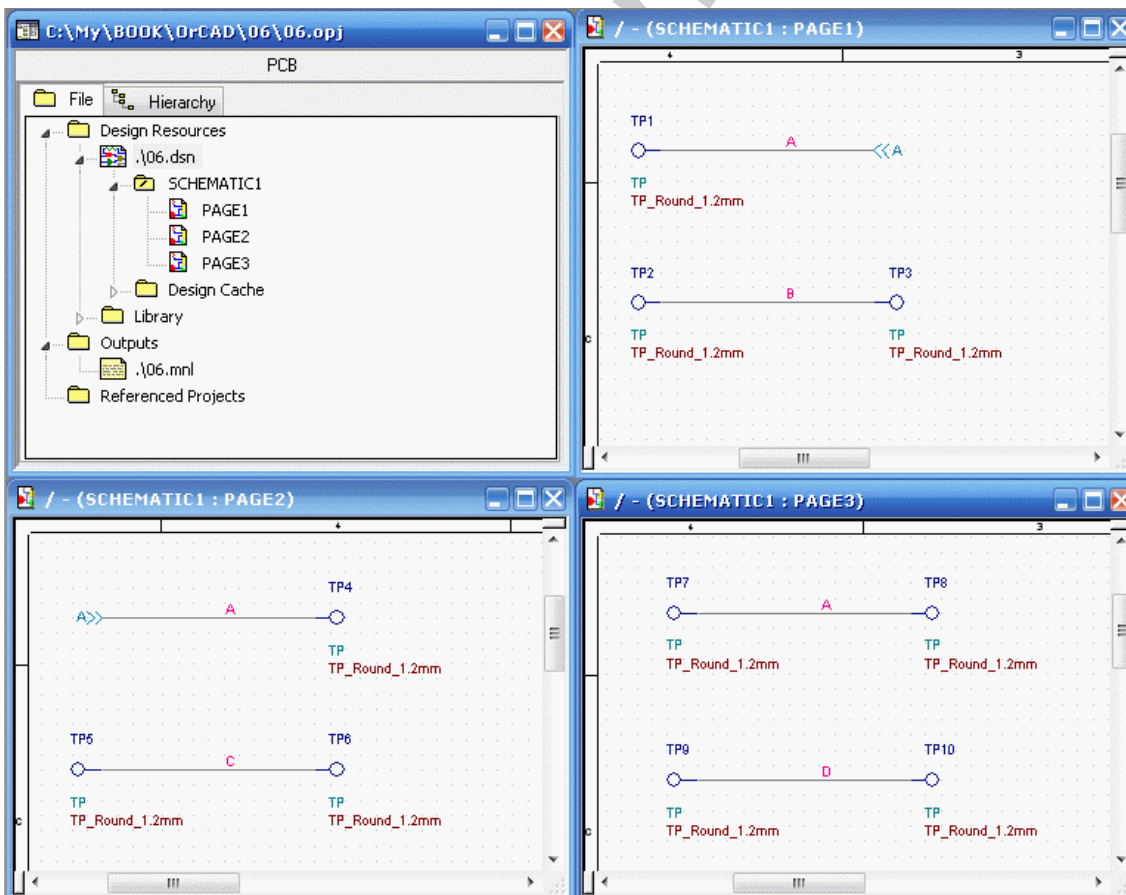


рис. 10-2

Список заданных цепей (**рис. 10-3**):

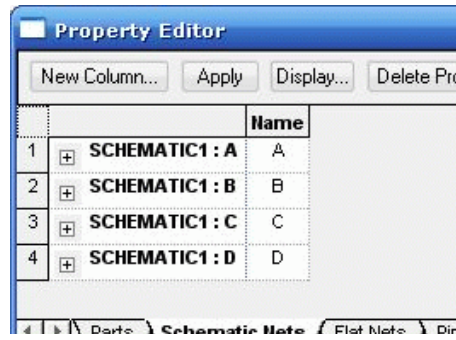


рис. 10-3

Реальный список цепей, Flat Nets (рис. 10-4):

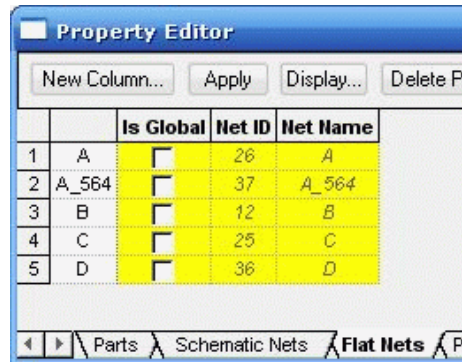


рис. 10-4

Изолированная цепь **A**, расположенная на последнем листе, получила уникальное имя **A_564**.

Файл Layout Plus, созданный на основе полученного netlist-a (рис. 10-5):

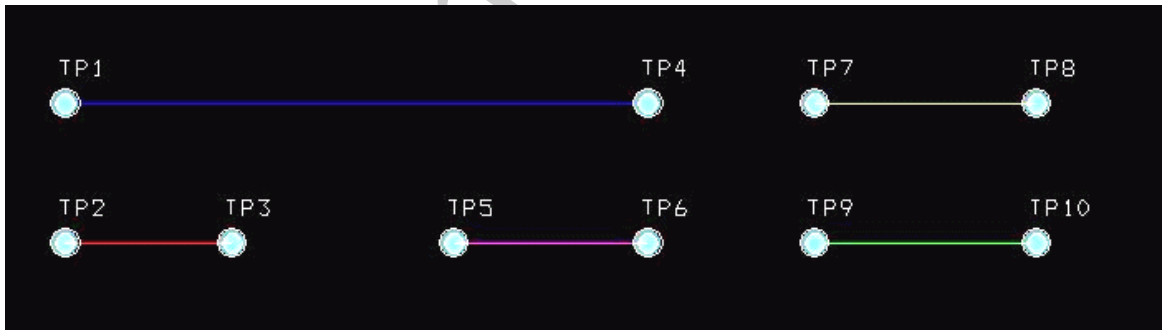


рис. 10-5

И, наконец, список цепей Layout (рис. 10-6):

Net Name	Color	Width Min Con Max	Routing Enabled	Share	Weight	Reconn Rule
A	Blue	12	Yes	Yes	50	Std
A_564	Yellow	12	Yes	Yes	50	Std
B	Red	12	Yes	Yes	50	Std
C	Magenta	12	Yes	Yes	50	Std
D	Green	12	Yes	Yes	50	Std

рис. 10-6

Если Вы внимательно читали предыдущий раздел, то ничего нового или неожиданного для себя в этой главе не обнаружили.

10.2. Simple hierarchies.

Если проектируемое устройство достаточно сложно, то удобно разбить схему на функциональные блоки. Вот как это делается.

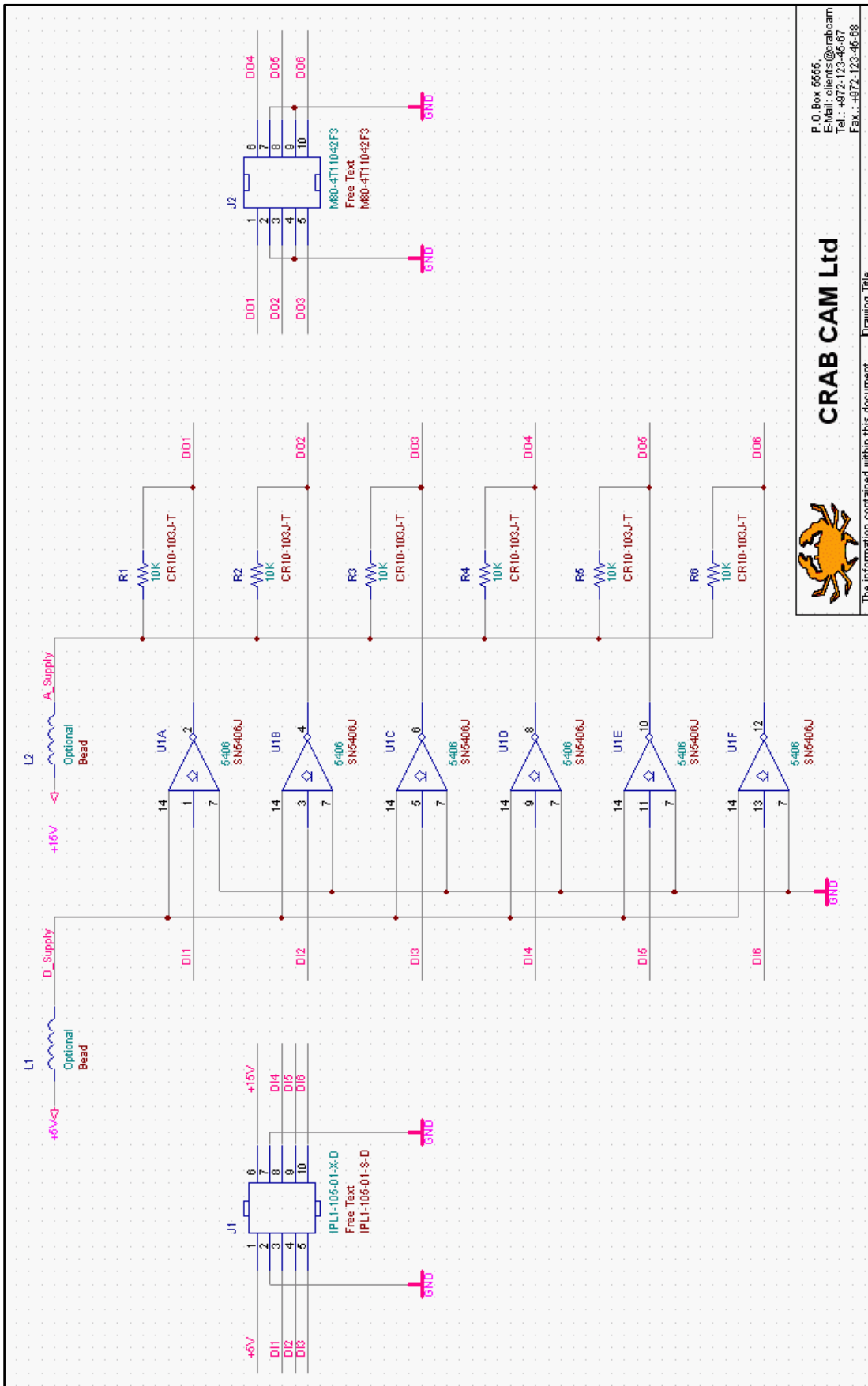


рис. 10-7

На **рис. 10-7** показана схема преобразователя цифрового сигнала TTL в сигнал с уровнем +15V. Устройство построено на м/с SN5406. Это – 6 инверторов с открытым коллектором.

На разъём **J1** подаётся питание +5V и +15V, а также входные данные. Разъём **J2** – выходной.

Вся схема уместится на одном листе (**рис. 10-8**).

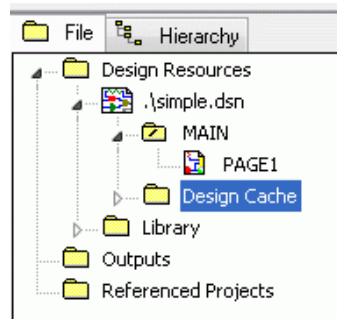


рис. 10-8

Давайте выделим микросхему и сопутствующие ей компоненты в отдельный функциональный блок. Для этого в Project Manager создайте новую папку – **New Schematic** (**рис. 10-9**), которую назовите, к примеру, «**Buffer**». Затем добавьте в неё новый лист, как показано на **рис. 10-10**.

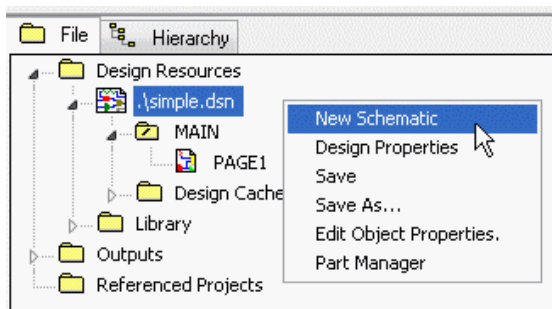


рис. 10-9

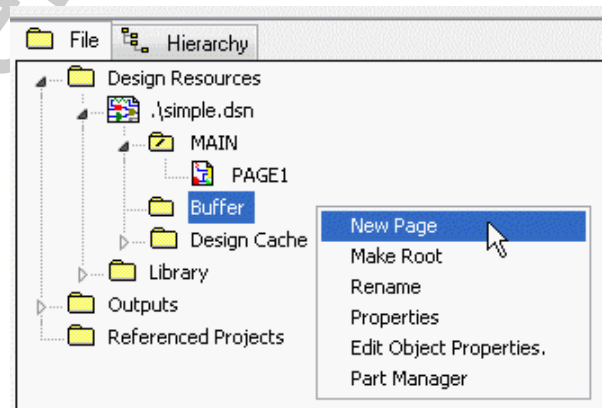


рис. 10-10

В результате Ваш проект должен выглядеть следующим образом (**рис. 10-11**):

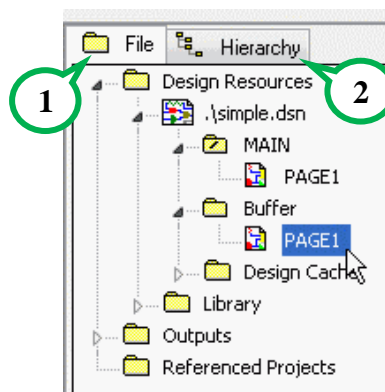
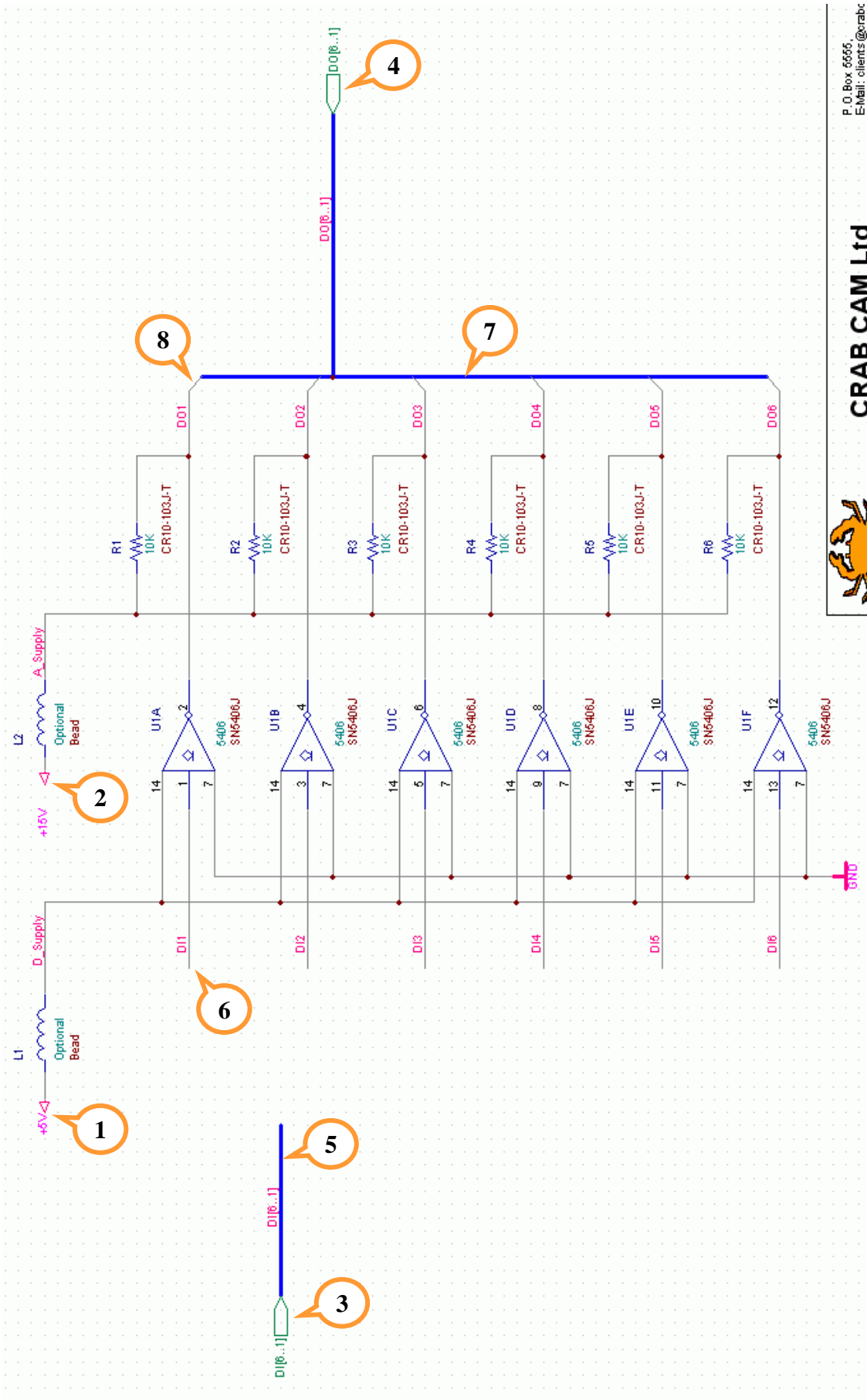


рис. 10-11

Выделите и перенесите необходимые элементы схемы на новый лист.

Если в пределах одной папки Schematic электрические цепи на разных листах соединяются между собой посредством Off-Page Connectors, то соединение между различными Schematics осуществляется с помощью соединителей **Port**.



P. O. Box 5555,
E-Mail: clients@crabc
Tel: +0775 173 64 67

CRAB CAM Ltd

рис. 10-12

Соединители Port, в отличие от Off-Page connectors, обладают свойством направленности **Type**: *Input*, *Output*, *Bidirectional*, *Passive* и т.д.. Если Вы определите тип порта неверно,

соединение не нарушится, но утилита DRC покажет ошибку.

Имеется шесть линий входа, которые называются **DI1, DI2 .. DI6** и шесть линий выхода: **DO1, DO2 .. DO6**. Чтобы не создавать 12 портов, разумно объединить их в шины. На **рис. 10-12** показано, как это сделать.

Выходные линии образуют шину **DO[6..1]** (7), которая оканчивается портом типа Output (4). Показан вариант подключения выводов к шине с помощью элементов **Bus Entry** (8).

Линии входа тоже объединены в шину с именем **DI[6..1]** (5). Шина начинается Input-портом (3) и не имеет визуального подключения к выводам (6).

Оба показанных способа абсолютно правильные.

Земля, а также питание +5V и +15V подключены к соединителям типа Global, поэтому нет необходимости создавать три дополнительных порта.

Теперь вернитесь на **PAGE1, Schematic MAIN**. На листе остались только входной и выходной коннекторы (**рис. 10-13**).

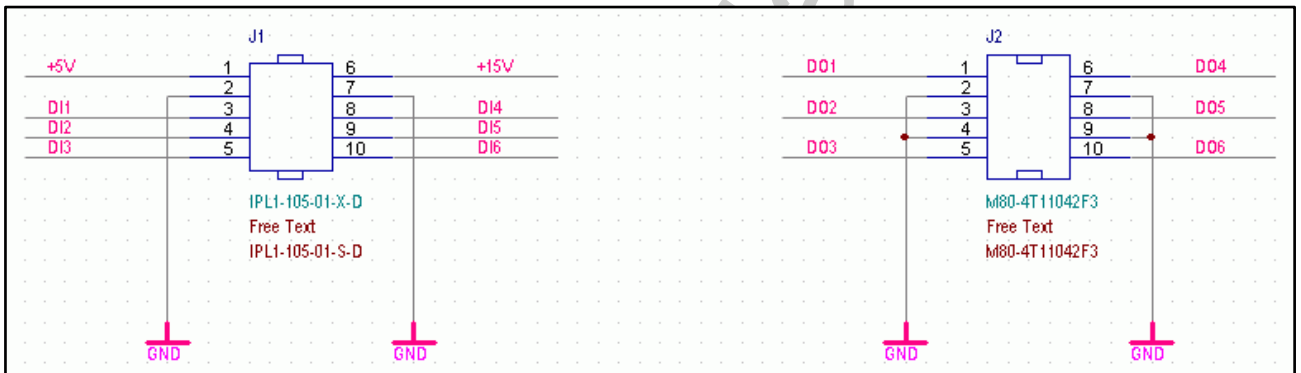


рис. 10-13

На этом листе мы размещаем **Hierarchical Block**. Выполните команду: «**Place** → **Hierarchical Block...**».

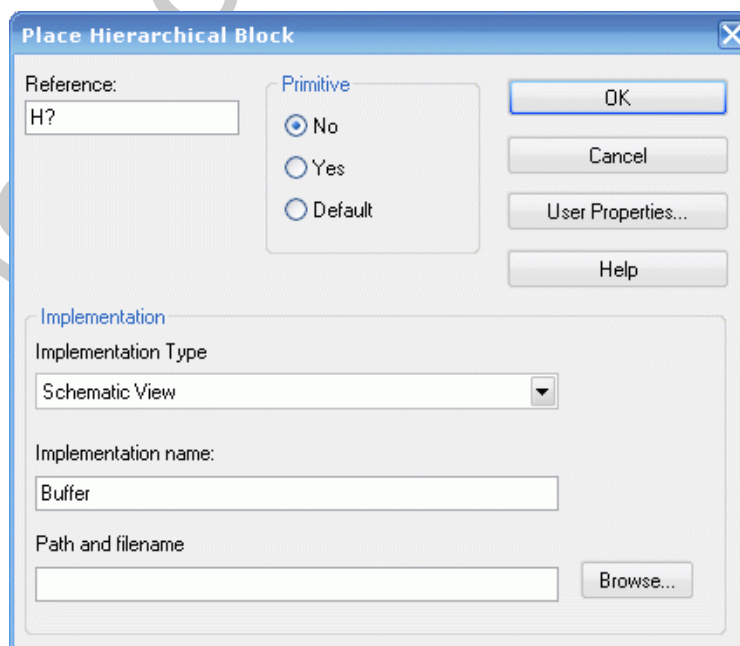


рис. 10-14

В появившемся окне заполните поля:

- Reference: **H?**. Иерархические блоки, как правило, обозначают буквой «H».

- Primitive: **No**. Прimitives называют компоненты, внутренняя структура которых нас не интересует или недоступна. Поскольку Buffer – это часть схемы, состоящая из многих компонентов, следовательно, примитивом она не является.
- Implementation Type: **Schematic View**.
- Implementation name: **Buffer**. В этом поле указываем название папки Schematic, которая представляет данный блок.

Нажмите <OK>. Теперь укажите мышкой прямоугольную область, в которой иерархический блок будет создан.

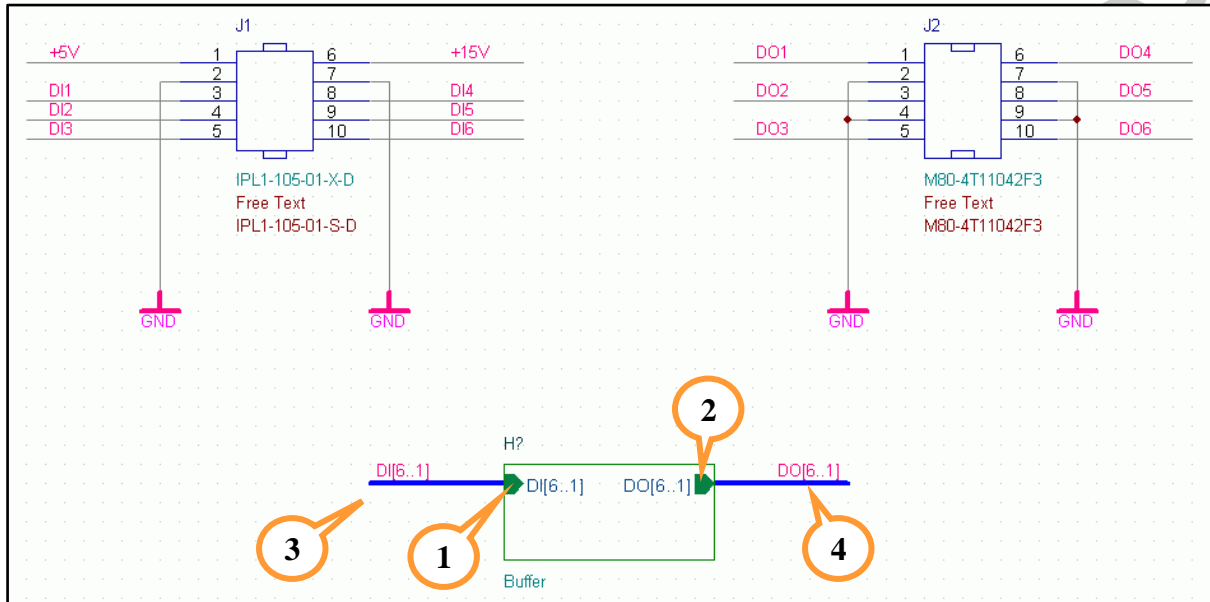


рис. 10-15

Иерархический блок, по аналогии с программированием, можно представить как подпрограмму, т.е. часть кода, который выполняет определённую функцию. Подпрограмма принимает и возвращает данные посредством локальных переменных. В иерархическом блоке аналогичной цели служат порты. На рис. 10-15 видно, что Capture автоматически создал два Hierarchical Pins: DI[6..1] и DO[6..1] (1, 2) в соответствии со схемой, показанной на рис. 10-12. Нам остаётся лишь подключить к ним одноимённые шины (3 и 4).

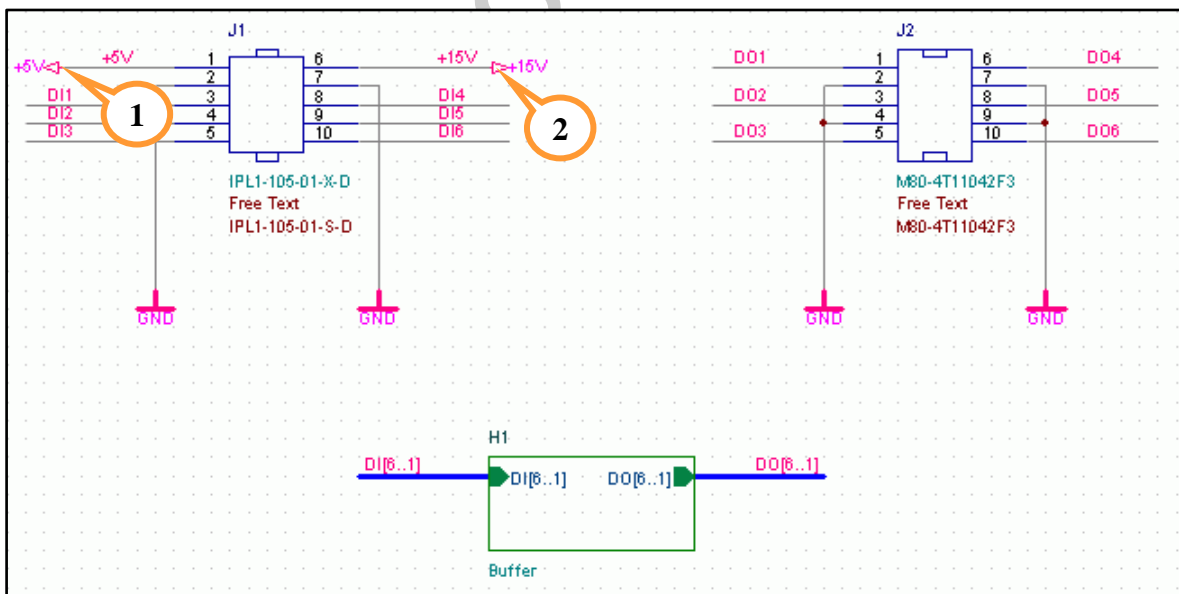


рис. 10-16

С помощью команды **Annotate** можно упорядочить нумерацию компонентов. Установите флаг **Include non-primitive parts**, чтобы пронумеровать по порядку имеющиеся в ди-

зайне иерархические блоки (рис. 10-17, рис. 10-16).

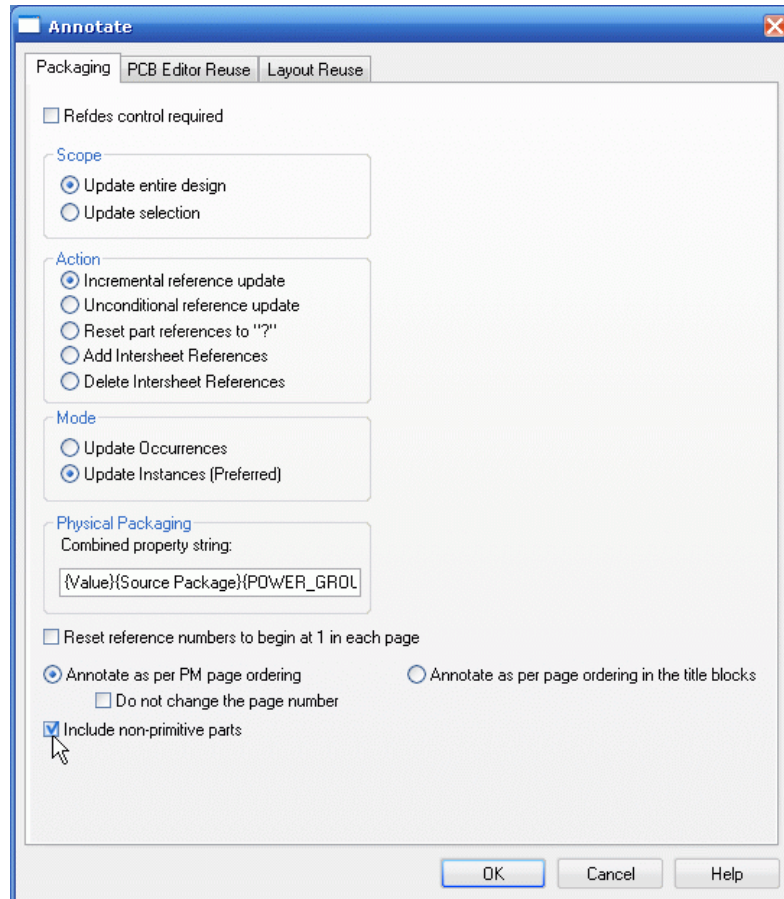


рис. 10-17

Не забудьте подключить VCC-соединители к цепям +5V и +15V, как показано на рис. 10-16 ① и ②.

Мы только что рассмотрели на примере, как можно разбить дизайн на функциональные блоки и оформить эти блоки в виде отдельных Schematics. Дизайн, состоящий из нескольких Schematic-папок, каждой из которых ставится в соответствие только один Hierarchical Block называется дизайном с простой иерархией (**Simply Hierarchy**). Говорят, что Schematics соединяются между собой вертикально. Если листы в пределах одной Schematic соединяются «горизонтально», то дизайн с иерархией можно назвать «трёхмерным».

10.3. Навигация по дизайну.

В одном дизайне может быть сколь угодно много Schematics, которые могут опускаться вниз на произвольное количество «этажей». Это значит, что любой блок можно разбить на подблоки. Таким образом, на самом верхнем уровне мы видим структурную или функциональную схему, а опускаясь ниже, получаем всё более детальное описание.

Папки Schematic мы видим в Project Manager (рис. 10-11), вкладка **Files** (①). Schematic самого верхнего уровня отображается первой и обозначается символом рис. 10-18.



рис. 10-18



рис. 10-19

Это, так называемая, **Root Schematic**, головная схема.

Остальные Schematics располагаются в Project Manager в алфавитном порядке ниже

Root и обозначаются символом **рис. 10-19**.

Во вкладке Hierarchy (**рис. 10-11**, ②), представлена зависимость Schematics. Например, из **рис. 10-20** ясно, что дизайн состоит из главной (Root) схемы **Main**. На схеме имеются три компонента: разъёмы **J1** и **J2** (Primitive) и иерархический модуль **H1**.

Напротив **H1** стоит стрелка (**рис. 10-20**, ①), следовательно, модуль является компонентом Non-Primitive, и нам доступна его внутренняя структура. Щёлкнув по стрелке, можно увидеть список компонентов или других модулей, входящих в **H1** (**рис. 10-21**).

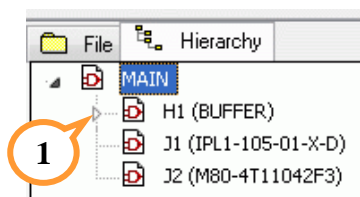


рис. 10-20

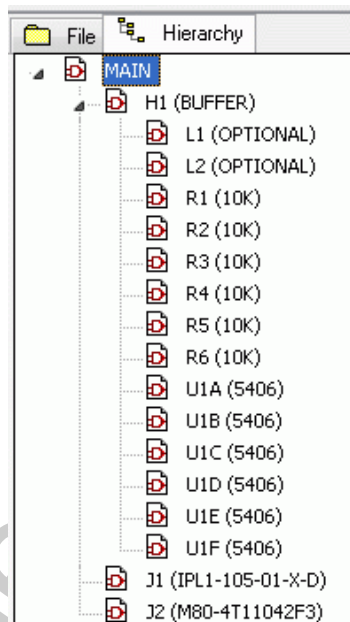


рис. 10-21

Все иерархические модули в конечном итоге должны иметь электрическое соединение с Root Schematic. Если это не так, то изолированный модуль будет исключён из netlist-а и из дерева Hierarchy.

Непосредственно на схеме движение с верхнего уровня на нижние осуществляется следующим образом (**рис. 10-22**):

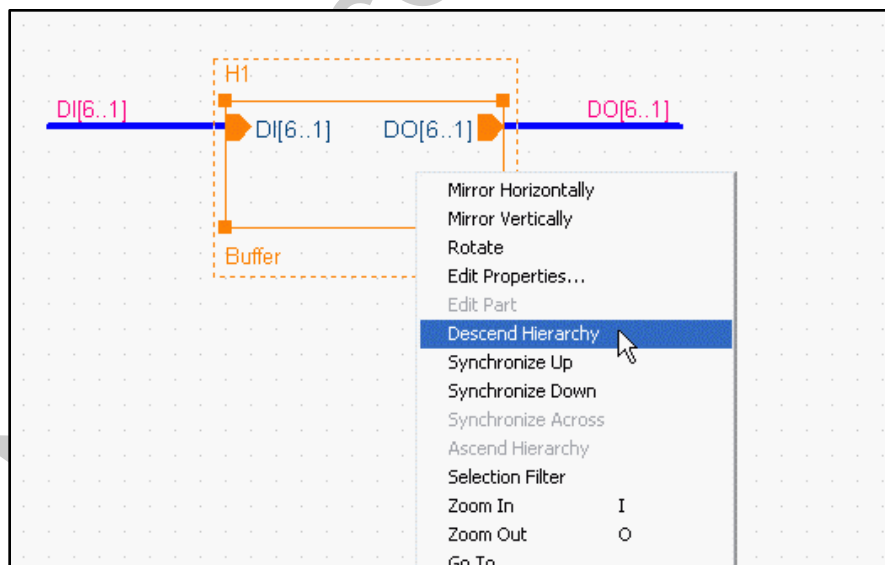


рис. 10-22

Выделите левой кнопкой мыши иерархический блок. Затем, щёлкнув правой кнопкой, выберите: **Descend Hierarchy**¹¹⁷.

¹¹⁷ В последних версиях OrCAD имеется возможность спуска на уровень вниз по двойному щелчку мыши.

Для движения вверх нужно в произвольном месте схемы щёлкнуть правой кнопкой мыши и выбрать: **Ascend Hierarchy** (рис. 10-23).

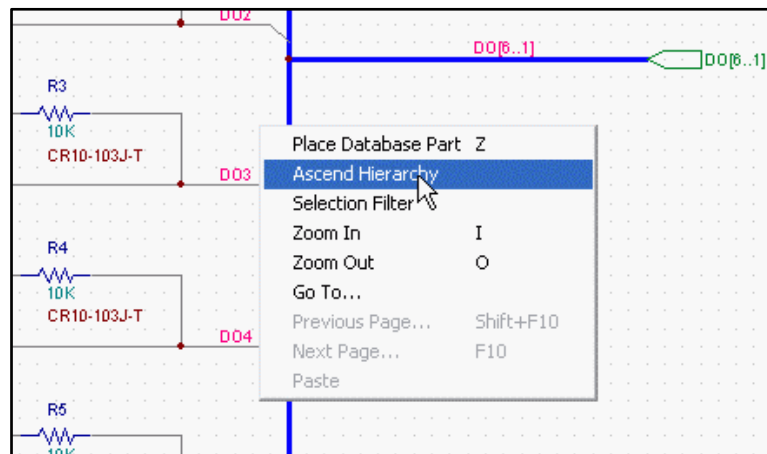


рис. 10-23

10.4. Synchronize connections.

Разберёмся, как вносить изменения в иерархические блоки.

Выполните проверку DRC для схемы, показанной на рис. 10-16. К большому удивлению, мы увидим предупреждение, что цепи питания ни к чему не подключены (рис. 10-24).

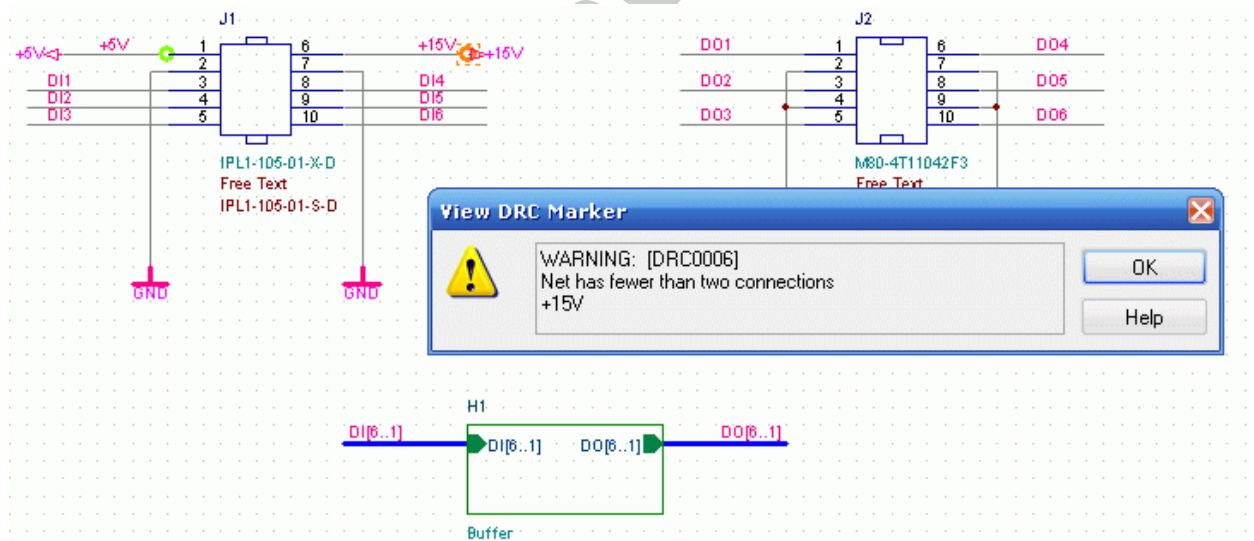


рис. 10-24

Это не так, на самом деле. Capture DRC не учитывает, что цепь соединена посредством Global connectors. На мой взгляд, это – недочёт программы. Про это надо просто знать.

На всякий случай, проверим, что соединение не нарушено. Для этого передайте netlist в Layout (рис. 10-25).

Как видите, всё в порядке. Красная ratsnest – это цепь +5V. Она соединяет L1 с первой ножкой J1. Сиреневая – цепь питания +15V. Соединяется L2 и шестая ножка коннектора.

Несмотря на предупреждение DRC, netlist формируется правильно. Тем не менее, давайте избавимся от этой «ошибки», для чего добавим в иерархический блок порты питания. Я рекомендую всегда делать так, чтобы, во-первых, избежать настоящей ошибки. Во-вторых, схема станет более наглядной и читаемой.

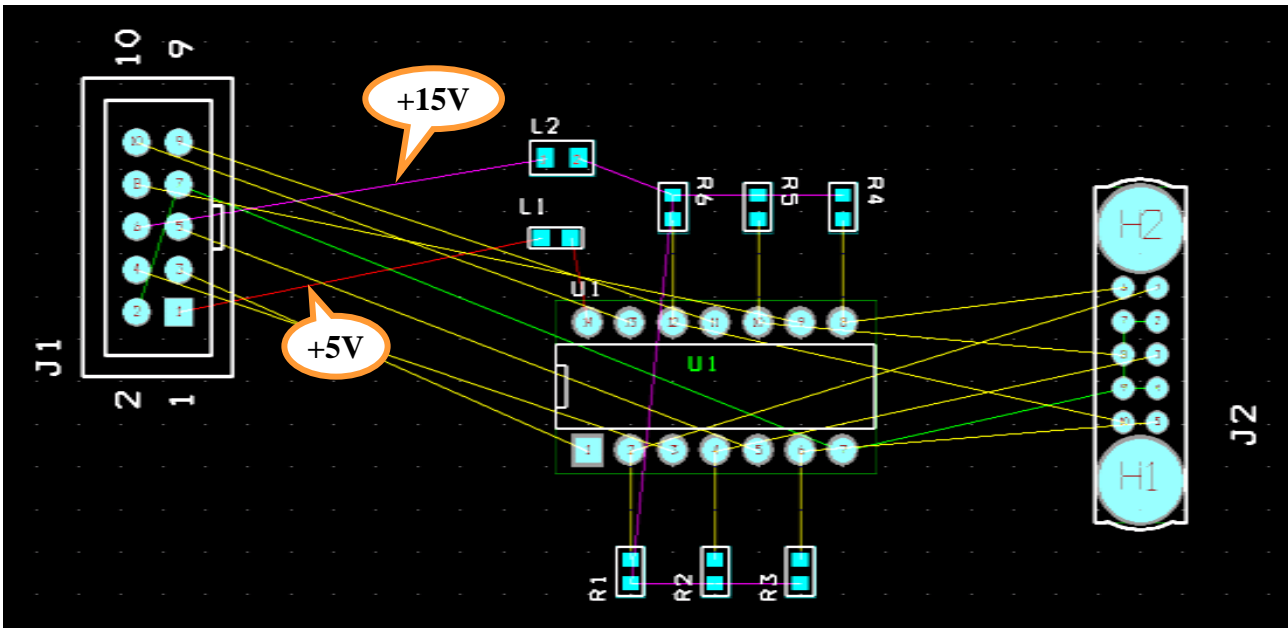


рис. 10-25

Сделать это можно двумя способами. Способ первый. Откройте схему **Buffer** и добавьте два порта типа **Power** (рис. 10-26, рис. 10-27). Пусть это будут **+15V** и **GND** (рис. 10-28).



рис. 10-26

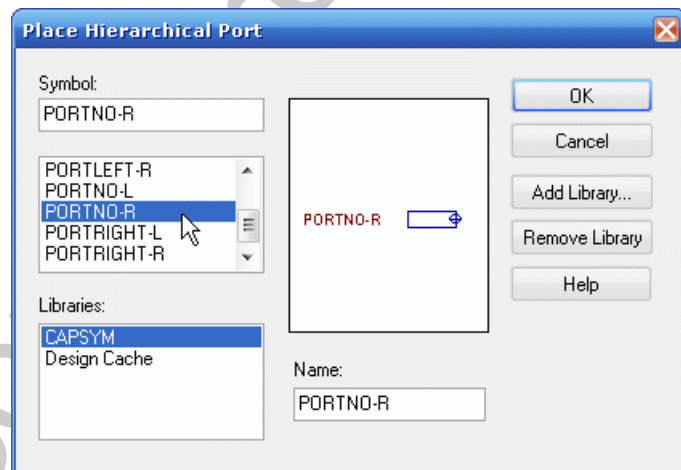


рис. 10-27

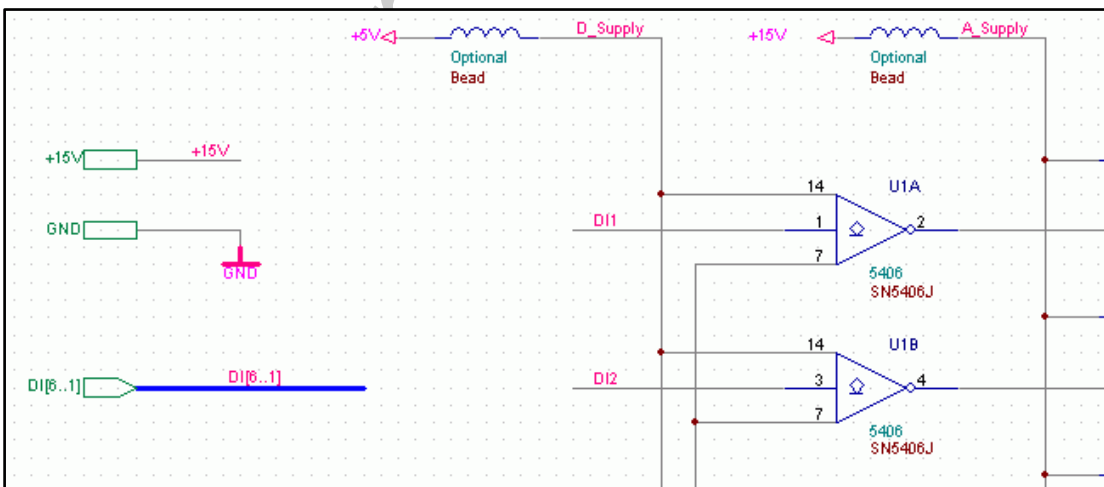


рис. 10-28

Поднимитесь на уровень выше (**Ascend Hierarchy**), выделите иерархический блок и из контекстного меню выполните команду **Synchronize Up** (рис. 10-29).

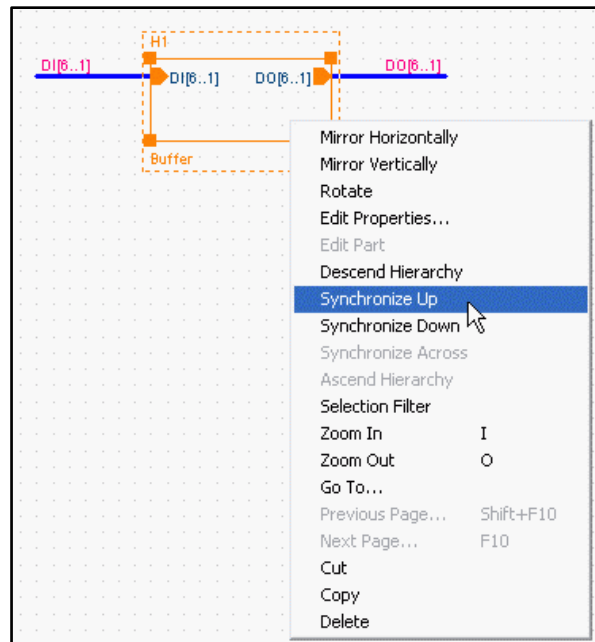


рис. 10-29

Вы увидите, что Capture добавил два дополнительных порта в блок.

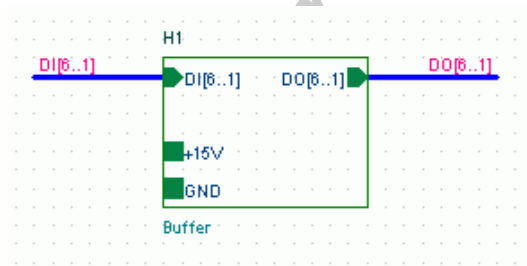


рис. 10-30

Способ второй. Выделите мышкой иерархический блок, затем с помощью инструмента **Place pin** (рис. 10-31) добавьте порт **+5V** типа **Power** (рис. 10-32).



рис. 10-31



рис. 10-32

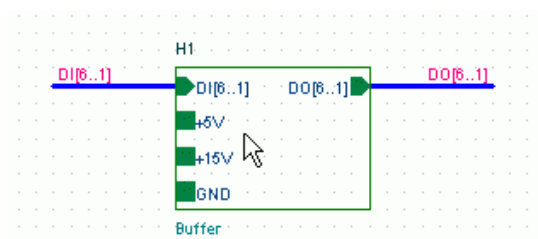


рис. 10-33

Вот что должно получиться – рис. 10-33.

Теперь, аналогично предыдущему случаю, выполните команду **Synchronize Down** (рис. 10-34).

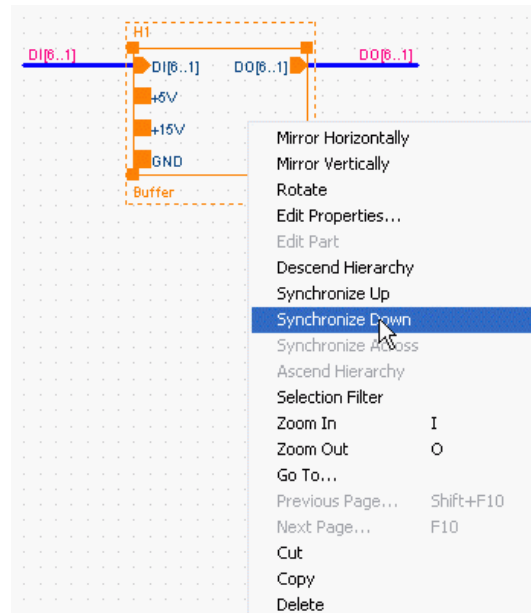


рис. 10-34

В **Buffer Schematic** будет добавлен соответствующий порт. Остаётся лишь правильно подключить его (рис. 10-35).

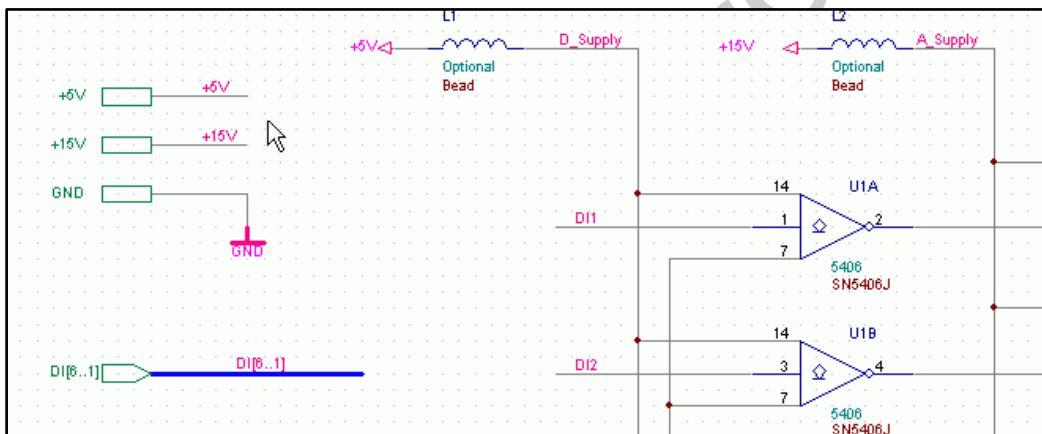


рис. 10-35

Не забудьте внести исправления в **Main Schematic** (рис. 10-36).

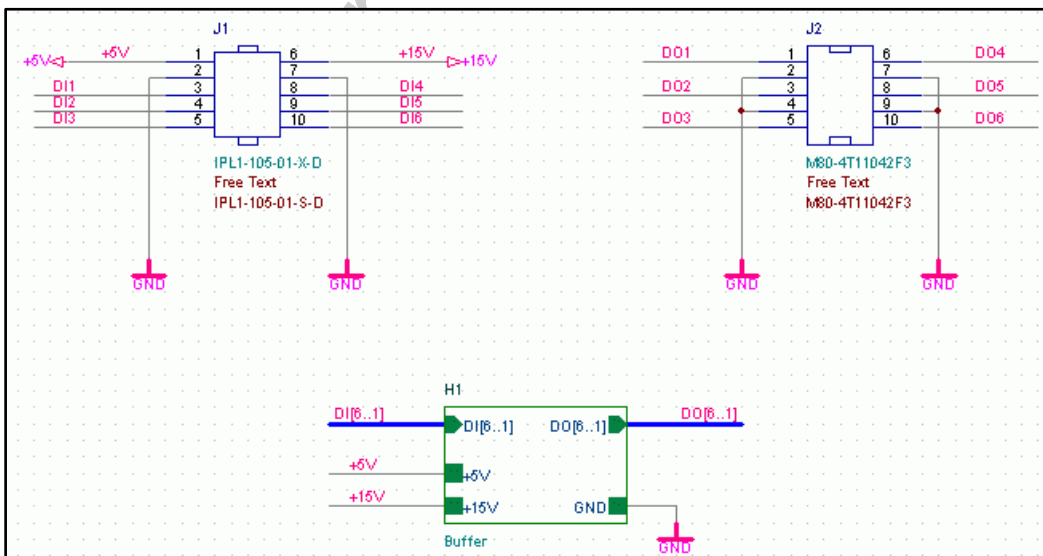


рис. 10-36

10.5. Complex hierarchies.

Если сравнивать иерархические блоки с подпрограммами, то, подобно подпрограммам, их можно использовать в дизайне произвольное количество раз. В этом случае говорят о дизайне со сложной иерархией (**Complex Hierarchy**).

Для того чтобы корректно провести следующий пример, немного измените схему **Buffer**, как показано на **рис. 10-37**. Измените все названия **+5V** на **VCCD**, а **+15V** – на **VCCB**.

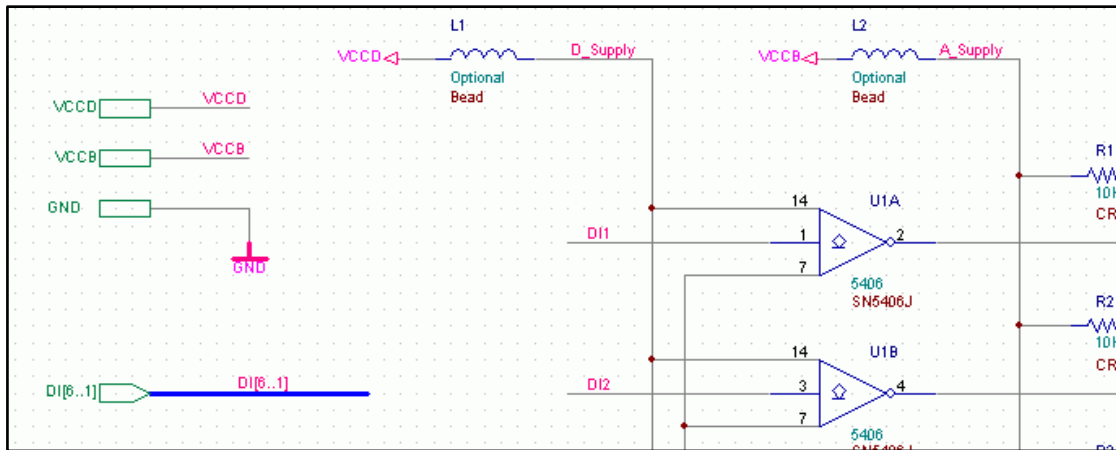


рис. 10-37

Соответственно, в схеме Main изменятся наименования выводов блока (**рис. 10-38**):

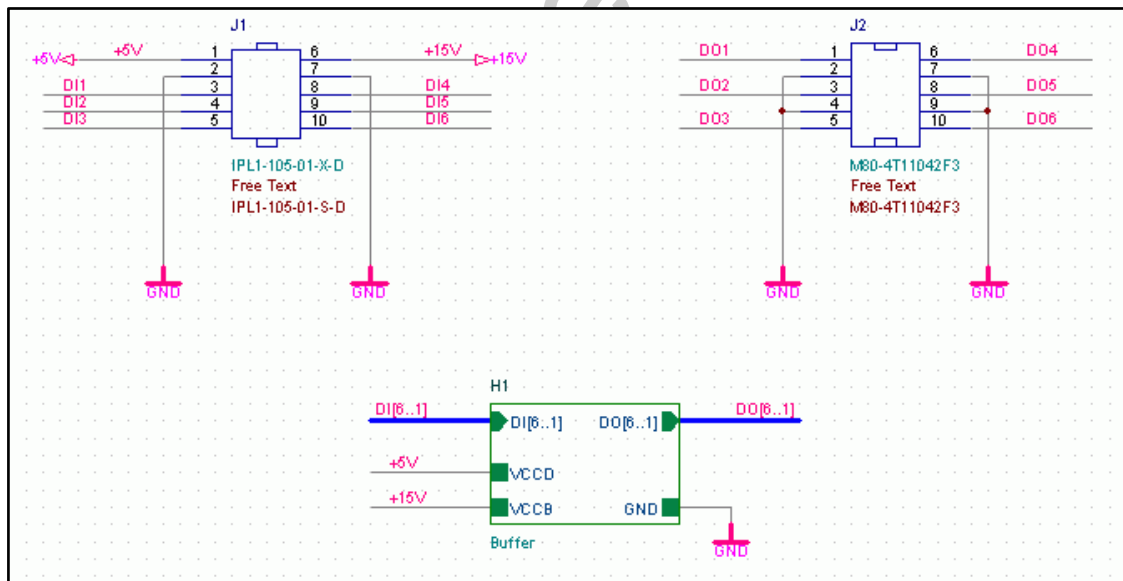


рис. 10-38

Теперь изменим условия. Поставим на схему два выходных разъёма. Один будет служить для подключения нагрузки на **+15V**, а другой – для **+5V**.

Посмотрите на **рис. 10-39**. Сначала скопируйте разъём **J2**. Затем переименуйте выводы у нового разъёма. Вместо **DO1 .. DO6** назовём их, скажем, **DT1 .. DT6**.

Теперь сделайте рядом копию иерархического блока. Входная шина остаётся без изменений, а выходные сигналы подключаем к новому разъёму. Для этого просто переименуем выходную шину из **DO[6..1]** в **DT[6..1]**. Порт **VCCB** подключим к цепи **+5V**.

Прежде, чем двигаться дальше, откройте окно Project Manager и выполните: **«Edit → Object Properties»** (**рис. 10-40**).

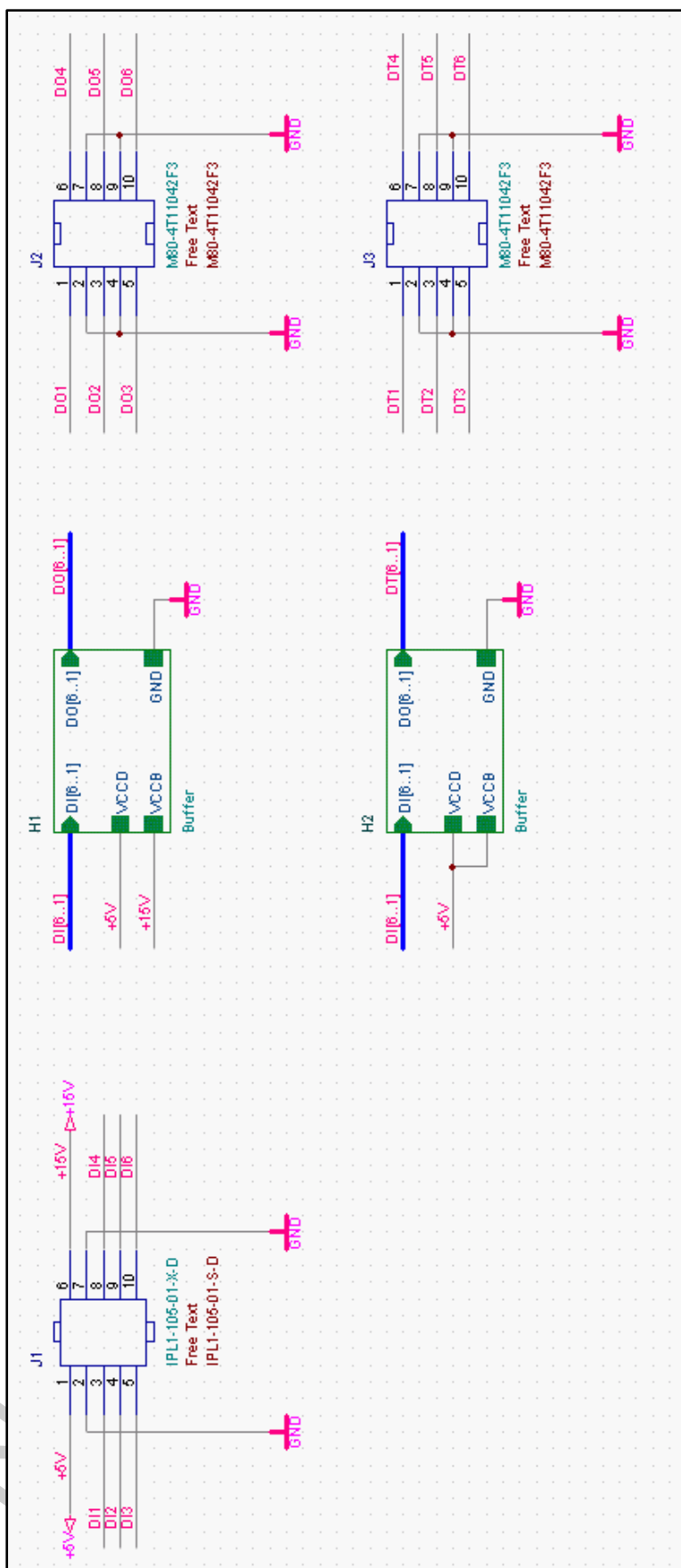


рис. 10-39

Property Editor покажет все компоненты дизайна. Мы видим, что на схеме присутствуют два блока **Buffer**. В то же время, микросхема **U1** только одна, что странно, поскольку понятно, что таких микросхем должно быть две. Равно как и всех остальных компонентов,

расположенных вместе с ней.

New Column... Apply Delete Property Filter by: OrCad-Capture Help

	Value	Reference	Designator	PCB Footprint	Power Pins Visible	Primitive	Source Library	Implementation Type	Implementation
1	Optional	L1		SML_0603	<input checked="" type="checkbox"/>	DEFAULT	C:\ACADENCES\SPB_16.0\TOOLS\CAPTURE\LIBRARY\DISCRE...	<none>	
2	Optional	L2		SML_0805	<input checked="" type="checkbox"/>	DEFAULT	C:\ACADENCES\SPB_16.0\TOOLS\CAPTURE\LIBRARY\DISCRE...	<none>	
3	10K	R1		SMR_0603	<input checked="" type="checkbox"/>	DEFAULT	C:\ACADENCES\SPB_16.0\TOOLS\CAPTURE\LIBRARY\ALTERA...	<none>	
4	10K	R2		SMR_0603	<input checked="" type="checkbox"/>	DEFAULT	C:\ACADENCES\SPB_16.0\TOOLS\CAPTURE\LIBRARY\ALTERA...	<none>	
5	10K	R3		SMR_0603	<input checked="" type="checkbox"/>	DEFAULT	C:\ACADENCES\SPB_16.0\TOOLS\CAPTURE\LIBRARY\ALTERA...	<none>	
6	10K	R4		SMR_0603	<input checked="" type="checkbox"/>	DEFAULT	C:\ACADENCES\SPB_16.0\TOOLS\CAPTURE\LIBRARY\ALTERA...	<none>	
7	10K	R5		SMR_0603	<input checked="" type="checkbox"/>	DEFAULT	C:\ACADENCES\SPB_16.0\TOOLS\CAPTURE\LIBRARY\ALTERA...	<none>	
8	10K	R6		SMR_0603	<input checked="" type="checkbox"/>	DEFAULT	C:\ACADENCES\SPB_16.0\TOOLS\CAPTURE\LIBRARY\ALTERA...	<none>	
9	5406	U1	C	DIP.100M14WV.3	<input checked="" type="checkbox"/>	DEFAULT	C:\ACADENCES\SPB_16.0\TOOLS\CAPTURE\LIBRARY\ALTERA...	<none>	
10	5406	U1	D	DIP.100M14WV.3	<input checked="" type="checkbox"/>	DEFAULT	C:\ACADENCES\SPB_16.0\TOOLS\CAPTURE\LIBRARY\ALTERA...	<none>	
11	5406	U1	A	DIP.100M14WV.3	<input checked="" type="checkbox"/>	DEFAULT	C:\ACADENCES\SPB_16.0\TOOLS\CAPTURE\LIBRARY\ALTERA...	<none>	
12	5406	U1	F	DIP.100M14WV.3	<input checked="" type="checkbox"/>	DEFAULT	C:\ACADENCES\SPB_16.0\TOOLS\CAPTURE\LIBRARY\ALTERA...	<none>	
13	5406	U1	E	DIP.100M14WV.3	<input checked="" type="checkbox"/>	DEFAULT	C:\ACADENCES\SPB_16.0\TOOLS\CAPTURE\LIBRARY\ALTERA...	<none>	
14	5406	U1	B	DIP.100M14WV.3	<input checked="" type="checkbox"/>	DEFAULT	C:\ACADENCES\SPB_16.0\TOOLS\CAPTURE\LIBRARY\ALTERA...	<none>	
15	Buffer	H1			<input checked="" type="checkbox"/>	NO		Schematic View	Buffer
16	Buffer	H2			<input checked="" type="checkbox"/>	NO		Schematic View	Buffer
17	PL1-105	J1		PL1-105-02	<input checked="" type="checkbox"/>	DEFAULT	C:\ACADENCES\SPB_16.0\TOOLS\CAPTURE\LIBRARY\ALTERA...	<none>	
18	M80-4T1	J2		M80-4T110	<input checked="" type="checkbox"/>	DEFAULT	C:\ACADENCES\SPB_16.0\TOOLS\CAPTURE\LIBRARY\ALTERA...	<none>	
19	M80-4T1	J3		M80-4T110	<input checked="" type="checkbox"/>	DEFAULT	C:\ACADENCES\SPB_16.0\TOOLS\CAPTURE\LIBRARY\ALTERA...	<none>	

рис. 10-40

Закроем **Property Editor** и выполним автоматическую нумерацию компонентов.

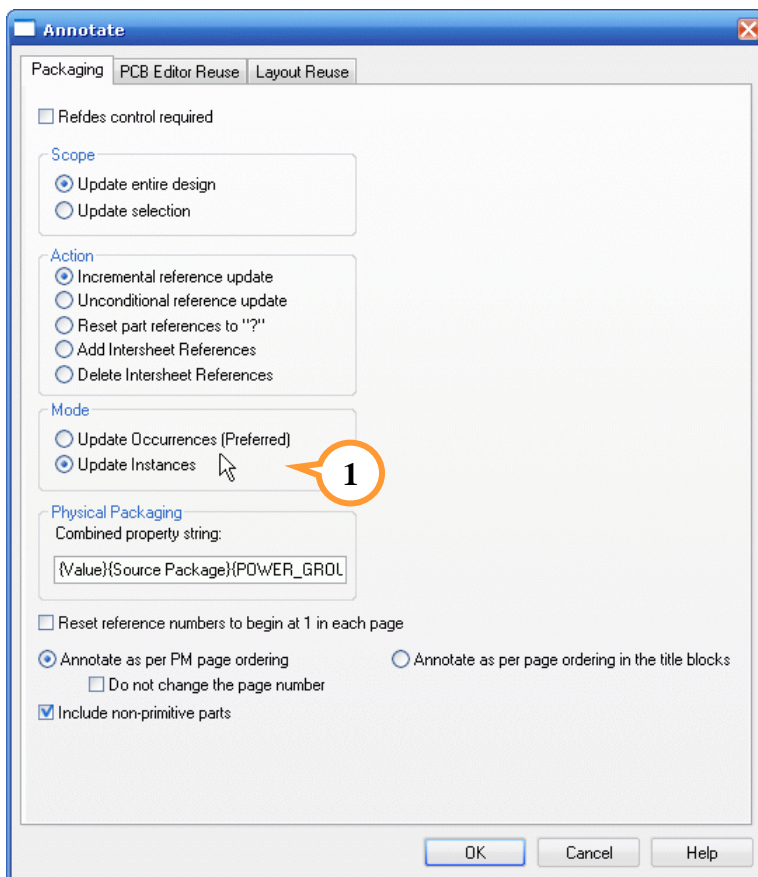


рис. 10-41

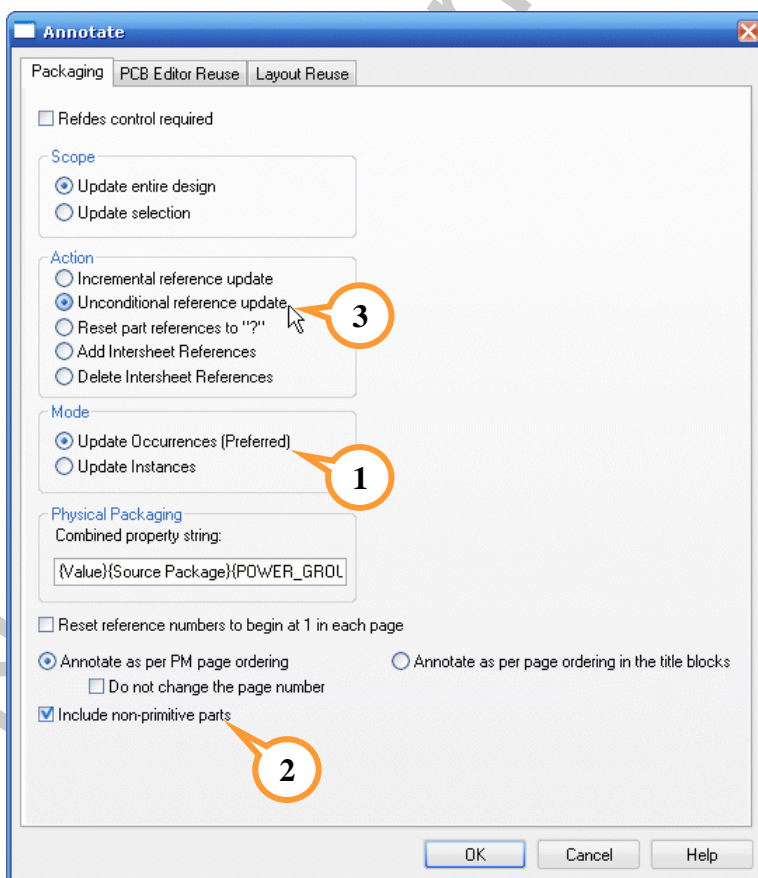


рис. 10-42

Когда откроется окно **Annotate**, обратите внимание на поле **Mode** (рис. 10-41, ①). Сэр-

ture указывает, что структура дизайна изменилась и требует сменить режим **Instances** на **Occurrences**. Что это значит?

Если Вы размещаете какой-либо компонент во Flat-дизайне или дизайне с простой иерархией, на схеме создаётся его **Instance** – копия компонента, взятая из соответствующей библиотеки. Если дизайн построен по принципу сложной иерархии, то могут создаваться копии Instance-ов - **Occurrences**. Таким образом, можно построить какой-либо модуль лишь однажды, а потом использовать его в дизайне произвольное количество раз.

Установите флаги, как показано на **рис. 10-42: Mode** → **Update Occurrences** (①), **Include non-primitive parts** (②), **Action** → **Unconditional reference update** (③) и нажмите <OK>. Посмотрим, что получилось.

MAIN Schematic не изменилась (**рис. 10-39**).

Войдите в блок **H1**. Здесь тоже изменений нет.

Теперь войдите в **H2**. А вот тут нас ожидает сюрприз! У компонентов другая нумерация! Наконец-то появилась микросхема **U2**! Да и остальные компоненты получили новые References.

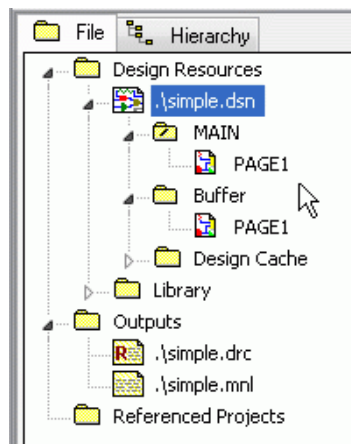


рис. 10-43

Вот как выглядит окно **Project Manager** (**рис. 10-43**). Как видно, существует единственная schematic **Buffer**. Однако, при попытке открыть её, Вы увидите окно выбора **Select Occurrence** (**рис. 10-45**).

На **рис. 10-44** показана иерархия дизайна. Здесь схема **Buffer** представлена дважды: блок **H1** и блок **H2**.

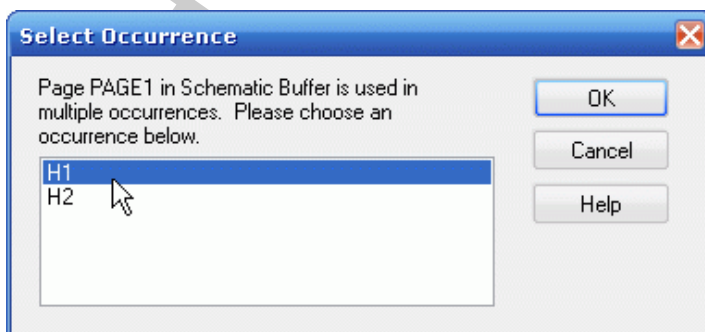


рис. 10-45

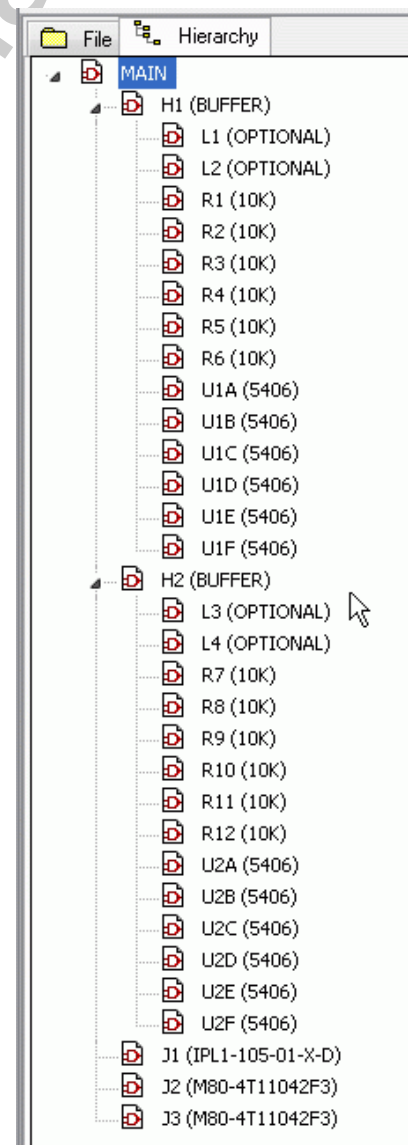


рис. 10-44

Откроем **Property Editor**, вкладку **Parts** (**рис. 10-46**).

	Value	Reference	Designator	PCB Footprint	Power Pins Visible	Primitive	Sou
1	Buffer : PAGE1 : L1	Optional	L1	SML_0603	<input checked="" type="checkbox"/>	DEFAULT	C:\CADENCE\SPB_16.0\TOO
2	H1.L1	Optional	L1	SML_0603	<input checked="" type="checkbox"/>	DEFAULT	C:\CADENCE\SPB_16.0\TOO
3	H2.L3	Optional	L3	SML_0603	<input checked="" type="checkbox"/>	DEFAULT	C:\CADENCE\SPB_16.0\TOO
4	Buffer : PAGE1 : L2	Optional	L2	SML_0805	<input checked="" type="checkbox"/>	DEFAULT	C:\CADENCE\SPB_16.0\TOO
5	H1.L2	Optional	L2	SML_0805	<input checked="" type="checkbox"/>	DEFAULT	C:\CADENCE\SPB_16.0\TOO

рис. 10-46

У instance **L1** есть 2 occurrences: **L1** и **L3**. Свойства occurrences приводятся в строках, выделенных жёлтым цветом. Фактически, это означает два компонента, присутствующие на схеме. Причём, свойства их различны. Например, одна из индуктивностей, расположенная в блоке **H1** имеет Reference **L1**, а другая – **L3**.

Абсолютно твёрдое правило: если в свойствах компонента указаны его occurrences (несколько или только один), то в расчёт принимаются именно occurrence properties. Иными словами: смотрим только на жёлтые строки, игнорируя белые!

39	H2.U2	5406	U2	E	DIP.100/14/W.3	<input checked="" type="checkbox"/>	DEFAULT	C
40	Buffer : PAGE1 : U1	5406	U1	B	DIP.100/14/W.3	<input checked="" type="checkbox"/>	DEFAULT	C
41	H1.U1	5406	U1	B	DIP.100/14/W.3	<input checked="" type="checkbox"/>	DEFAULT	C
42	H2.U2	5406	U2	B	DIP.100/14/W.3	<input checked="" type="checkbox"/>	DEFAULT	C
43	MAIN : PAGE1 : H1	Buffer	H1				NO	
44	MAIN : PAGE1 : H2	Buffer	H2				NO	
45	MAIN : PAGE1 : J1	PL1-105	J1		IPL1-105-02	<input checked="" type="checkbox"/>	DEFAULT	C
46	MAIN : PAGE1 : J2	M80-4T1	J2		M80-4T110	<input checked="" type="checkbox"/>	DEFAULT	C
47	MAIN : PAGE1 : J3	M80-4T1	J3		M80-4T110	<input checked="" type="checkbox"/>	DEFAULT	C

рис. 10-47

Иное дело – компоненты, не имеющие occurrences. Например, разъём **J1** (рис. 10-47). В данном случае, всё как обычно. Жёлтых строк нет, значит, смотрим на белую.

Правило «жёлтых строк» справедливо не только для реальных компонентов, но и для всех остальных элементов дизайна. Вот, например, Property Tab для **Title Blocks** (рис. 10-48):

	Page Number	Page Count	Schematic Name	Title	D
1	MAIN : PAGE1 : My Title Block	1	2	MAIN	
2	My Title Block	1	3	MAIN	
3	Buffer : PAGE1 : My Title Block	2	2	Buffer	
4	H1/My Title Block	2	3	Buffer	
5	H2/My Title Block	3	3	Buffer	

рис. 10-48

Если дизайн распечатать на бумаге, то реально мы получим 3 листа, поэтому общее количество страниц, указанное в столбце **Page Count** равно трём (①) (смотрим только на жёлтые строчки!). Номера страниц, указаны в поле **Page Number** (②).

На рис. 10-49 показан Title Block схемы **Main**. Номер листа (②) и общее количество листов (①) можно видеть в правом нижнем углу.


		CRAB CAM Ltd		P. O. Box 5555, E-Mail: clients@crabcam.com Tel.: +972-123-45-67 Fax.: +972-123-45-68	
The information contained within this document is the property of CRAB CAM Ltd. It shall not be reproduced, disclosed or used in whole or in part without written approval from CRAB CAM Ltd.			Drawing Title <div style="text-align: center; color: red; font-weight: bold;"> <Title1> MAIN </div>		
Design	Signature	Date	Size	Page Path	Drawing No.
	Kaikov D.M.	<Design Date>	A4	/	<Drawing No.>
Approved	<Approved By>	<Approved Date>	Monday, September 01, 2008 Thursday, December 11, 2008		Sheet 1 of 3

рис. 10-49

10.6. Цепи в дизайне со сложной иерархией.

Разобравшись с компонентами, проверим цепи.

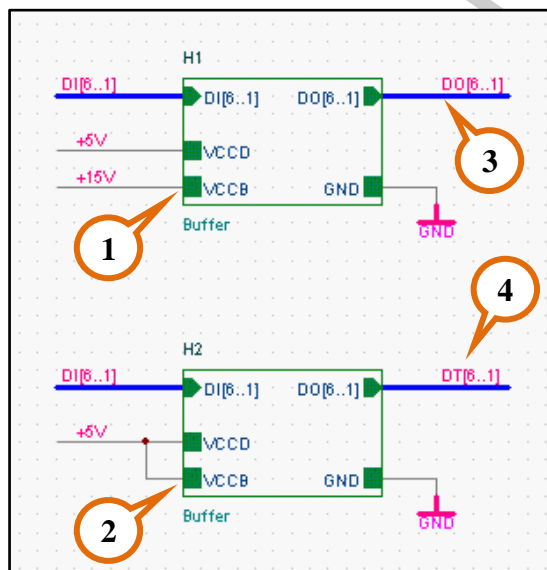


рис. 10-50

Напомню, как выглядит дизайн в настоящий момент.

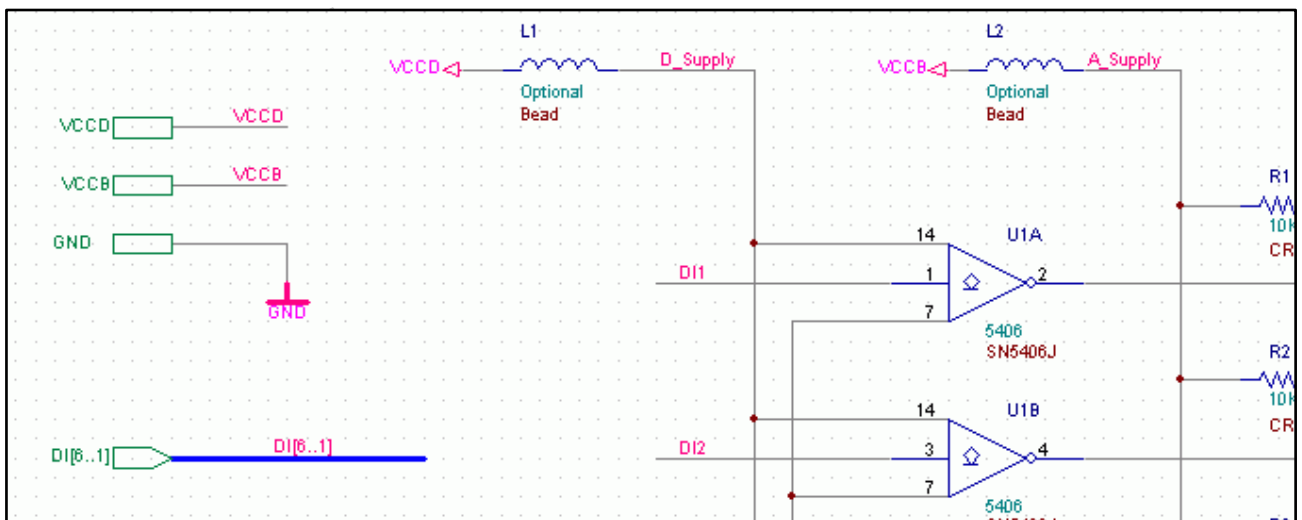


рис. 10-51

На схеме (рис. 10-50) имеется два одинаковых иерархических блока (рис. 10-51) (две подпрограммы). Разница заключается в том, что они получают разное напряжение питания (разные параметры). В первом случае, ко входу VCCB подключена цепь +15V (рис. 10-50,

①). Во втором случае – +5V (рис. 10-50, ②).

Соответственно, выход каждого из блоков «возвращает разные данные», и выходной порт DO[6..1] блока H1 подключён к шине DO[6..1] (рис. 10-50, ③), а блока H2 – к шине DT[6..1] (рис. 10-50, ④).

Откройте Property Editor, вкладку Schematic Nets (рис. 10-52).

		Name	Net Name
14	+	Buffer : DO6	DO6
15	+	Buffer : GND	GND
16	+	Buffer : VCCB	VCCB
17	+	Buffer : VCCD	VCCD
18	+	MAIN : +5V	+5V
19	+	MAIN : +15V	+15V
20	+	MAIN : DI1	DI1
21	+	MAIN : DI2	DI2
22	+	MAIN : DI3	DI3

рис. 10-52

		Name	Net Name
15	+	Buffer : GND	GND
16	-	Buffer : VCCB	VCCB
17	+	H1/VCCB	VCCB
18	+	H2/VCCB	VCCB
19	-	Buffer : VCCD	VCCD
20	+	H1/VCCD	VCCD
21	+	H2/VCCD	VCCD
22	-	MAIN : +5V	+5V
23	+	+5V	+5V
24	-	MAIN : +15V	+15V
25	+	+15V	+15V
26	+	MAIN : DI1	DI1
27	+	MAIN : DI2	DI2

рис. 10-53

Нам известно, что цепь VCCB одного из блоков подключена к цепи +15V, а другого – к цепи +5V, но из данного окна это определить невозможно. Окно Schematic Nets Properties выводит лишь список цепей, определённых в дизайне пользователем. Единственно, что можно установить, раскрыв свойства Occurrences, что цепей VCCB действительно две, и принадлежат они блокам H1 и H2 (рис. 10-53, ①).

Та же ситуация и с цепями A_SUPPLY (рис. 10-54). Всё, что нам известно – это то, что такие цепи есть в каждом из иерархических блоков. Соединяются ли они вместе или нет, сказать нельзя.

		Name	Net Name
1	-	Buffer : A_SUPPLY	A_SUPPLY
2	+	H1/A_SUPPLY	A_SUPPLY
3	+	H2/A_SUPPLY	A_SUPPLY
4	-	Buffer : D_SUPPLY	D_SUPPLY
5	+	H1/D_SUPPLY	D_SUPPLY
6	+	H2/D_SUPPLY	D_SUPPLY
7	+	Buffer : DI1	DI1
8	-	Buffer : DI2	DI2

рис. 10-54

		Is Global	Net ID	Net Name
1	+5V	<input checked="" type="checkbox"/>	104	+5V
2	+15V	<input checked="" type="checkbox"/>	419	+15V
3	A_SUPPLY	<input type="checkbox"/>	538	A_SUPPLY
4	A_SUPPLY_309	<input type="checkbox"/>	309	A_SUPPLY
5	D_SUPPLY	<input type="checkbox"/>	536	D_SUPPLY
6	D_SUPPLY_315	<input type="checkbox"/>	315	D_SUPPLY
7	DI1	<input type="checkbox"/>	102	DI1
8	DI2	<input type="checkbox"/>	96	DI2
9	DI3	<input type="checkbox"/>	98	DI3
10	DI4	<input type="checkbox"/>	113	DI4
11	DI5	<input type="checkbox"/>	112	DI5
12	DI6	<input type="checkbox"/>	105	DI6
13	DO1	<input type="checkbox"/>	153	DO1
14	DO2	<input type="checkbox"/>	154	DO2
15	DO3	<input type="checkbox"/>	152	DO3
16	DO4	<input type="checkbox"/>	155	DO4
17	DO5	<input type="checkbox"/>	156	DO5
18	DO6	<input type="checkbox"/>	151	DO6
19	DT1	<input type="checkbox"/>	503	DT1
20	DT2	<input type="checkbox"/>	501	DT2
21	DT3	<input type="checkbox"/>	500	DT3
22	DT4	<input type="checkbox"/>	499	DT4
23	DT5	<input type="checkbox"/>	498	DT5
24	DT6	<input type="checkbox"/>	502	DT6
25	GND	<input checked="" type="checkbox"/>	103	GND

рис. 10-55

Реальную картину можно увидеть, перейдя во вкладку Flat Nets (рис. 10-55).

Вот настоящие названия цепей, т.е. такие, как они будут отражены в Netlist: блок **H1** (рис. 10-56) и блок **H2** (рис. 10-57).

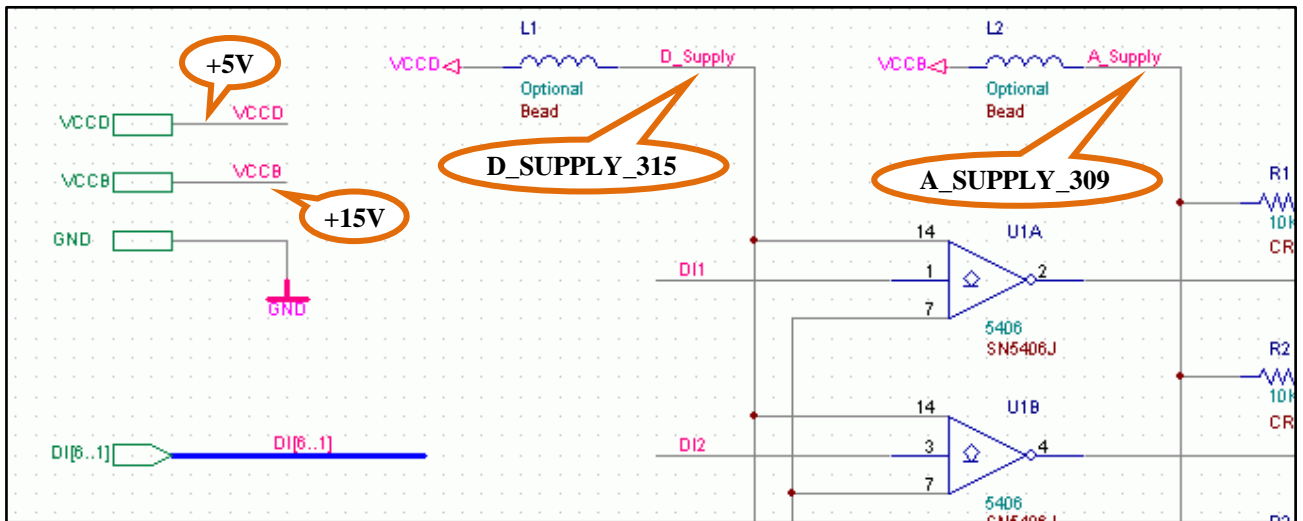


рис. 10-56

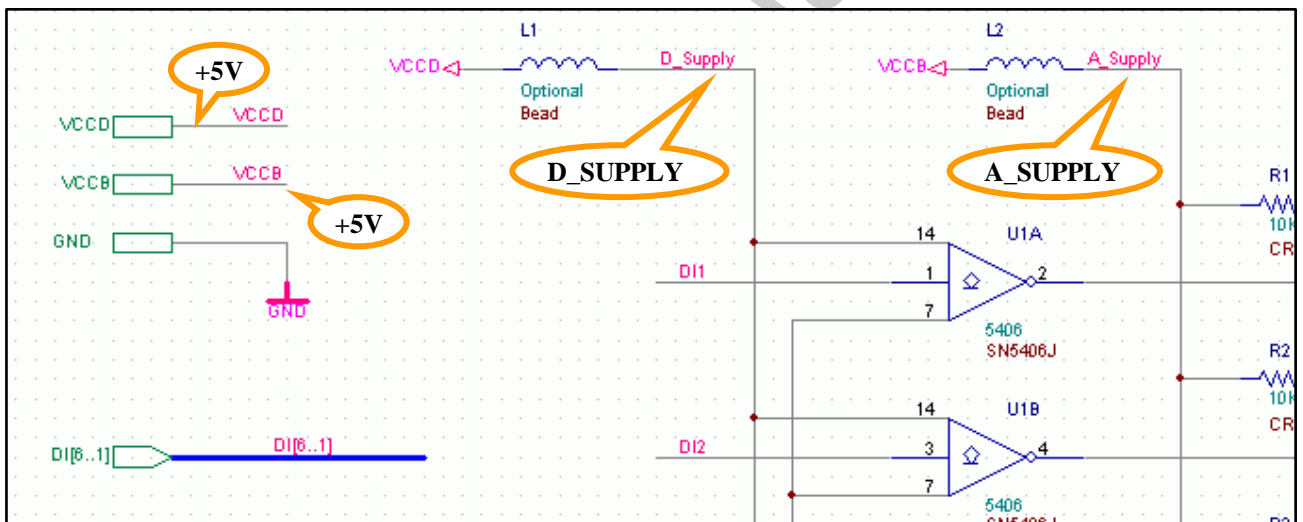


рис. 10-57

Подумайте, как это можно проверить¹¹⁸.

10.7. Библиотечный символ из иерархического блока.

Иерархическими блоками удобно пользоваться по многим причинам. Во-первых, схема разрабатываемого устройства предстаёт в виде совокупности функциональных модулей. Во-вторых, нет необходимости перерисовывать повторяющиеся участки, если такие имеются.

Однако все блоки выглядят одинаково, в виде безликих прямоугольников с выводами, что затрудняет восприятие схемы. Нельзя ли отредактировать внешний вид каждого блока в соответствии с его назначением?

Оказывается можно, и процесс этот совсем не отличается от создания обычного элемента.

Перейдите в Project Manager и выберите библиотеку, в которой будет размещаться символ иерархического блока. Затем создайте: «New Part» (рис. 10-58).

¹¹⁸Вы можете выделить цепь непосредственно на схеме и открыть её свойства Flat Net.

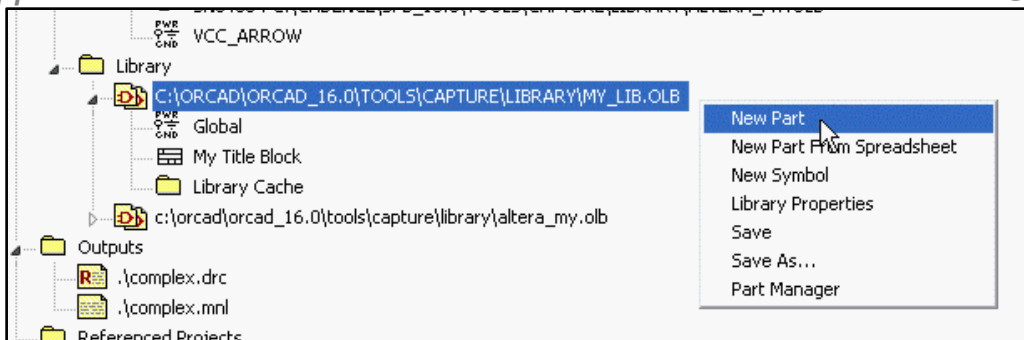


рис. 10-58

Укажите имя нового элемента (например, **Level Shifter**) и выберите букву, которой он будет обозначаться на принципиальной схеме (рис. 10-59). Поскольку это – иерархический блок, выберем букву “H”. Флажок **Pin Number Visible** можно снять, поскольку мы не создаём физический компонент и, следовательно, ножек у него нет.

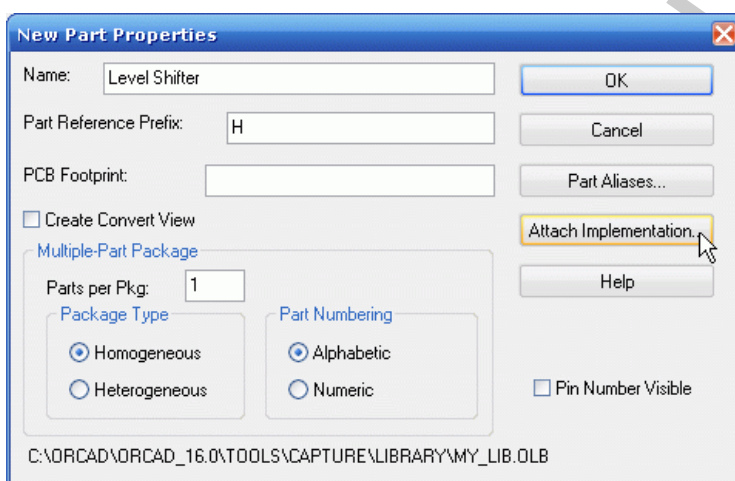


рис. 10-59

Далее, нажмите **<Attach Implementation>**.

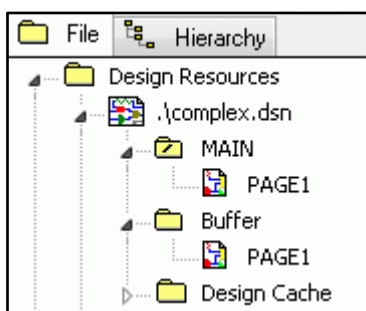


рис. 10-60

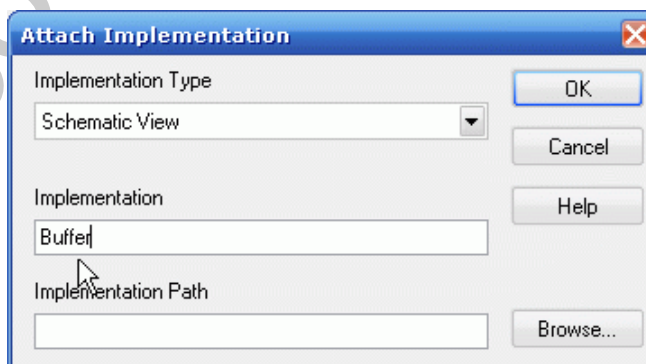


рис. 10-61

В поле **Implementation Type** (рис. 10-61) следует выбрать **Schematic View**, поскольку создаваемый элемент представляет, по сути, часть схемы.

В поле **Implementation** введите имя папки Schematic на которую будет ссылаться наш элемент. В данном случае – **Buffer** (рис. 10-60).

Нажмите **<OK>**, и можно приступать к рисованию. Я нарисовал вот такой символ (рис. 10-62):

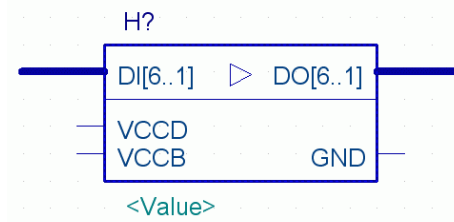


рис. 10-62

Вывод **DI[6..1]** типа **Input, Line, Bus**.

DO[6..1] – **Output, Line, Bus**;

VCCD, VCCB, GND – **Power, Short, Scalar**.

Как видно, выходы не имеют нумерации.

Сохраните полученный символ.

Теперь откройте нашу схему и замените один из иерархических блоков новым символом (рис. 10-63):

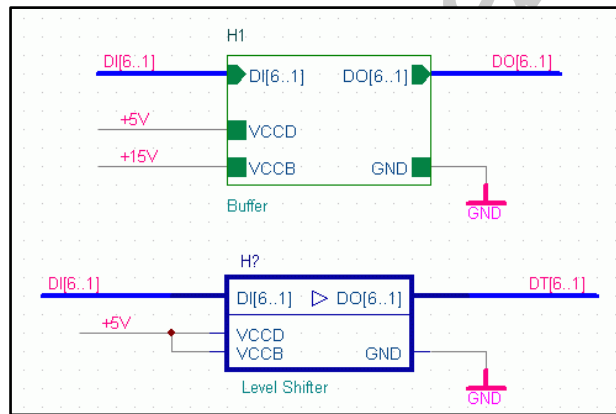


рис. 10-63

Обратите внимание, что при выборе символа из библиотеки, под его изображением появляется особая пиктограмма, указывающая, что элемент является иерархическим.

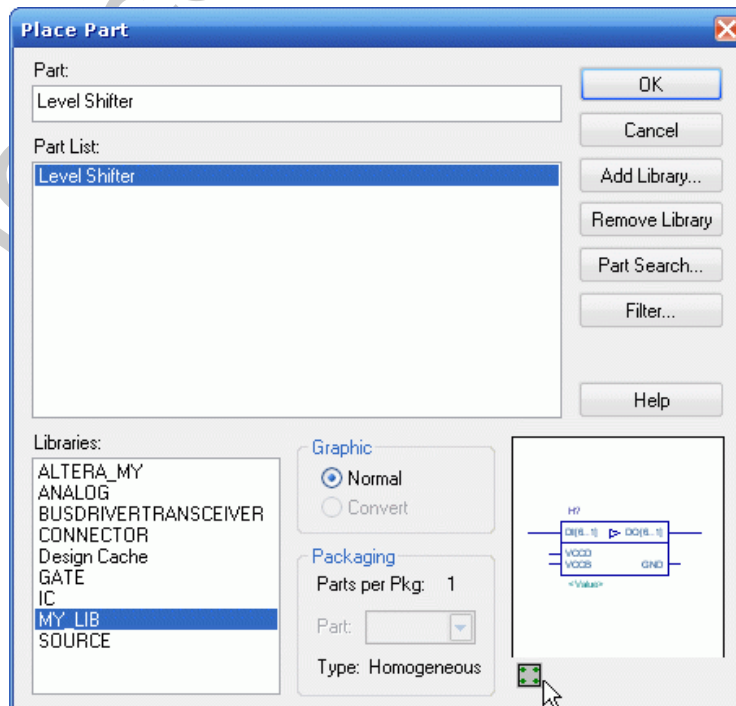


рис. 10-64

Осталась совсем малость: нужно определить элемент на схеме как **Nonprimitive** (рис. 10-65), чтобы нам была доступна его внутренняя структура.

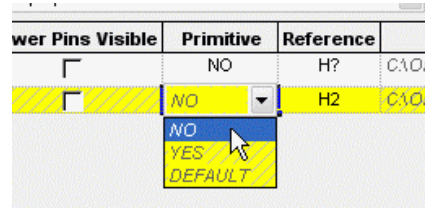


рис. 10-65

Пронумеруйте все элементы схемы с помощью команды Annotate, выполните проверку DRC и можете посмотреть дерево иерархии проекта (

рис. 10-66).

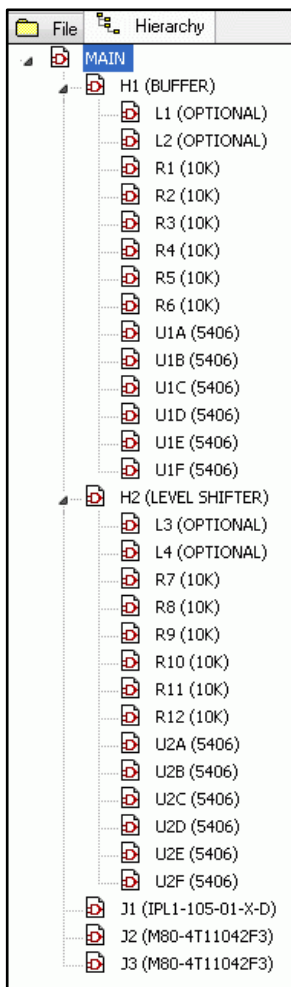


рис. 10-66

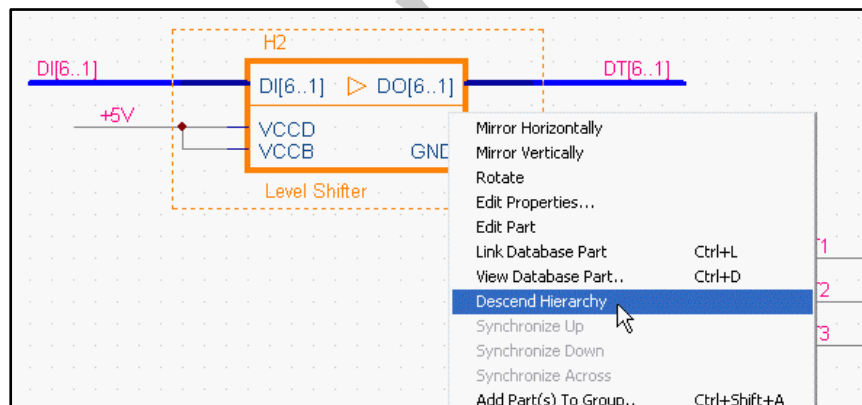


рис. 10-67

Структуру **H2** можно посмотреть, как обычно, командой **Descent Hierarchy** (рис. 10-67).

Иерархический блок **H1**, конечно, тоже можно заменить на **Level Shifter**.

10.8. Составной проект.

Ещё одно применение иерархических блоков: Вы можете «собрать» проект из разных файлов Capture. В компании может быть несколько человек, каждый из которых работает над своей частью задания. Например, один человек создаёт источник питания, ещё один работает над какой-то аналоговой частью, кто-то рисует схему управления, а кто-то занимается со-

единением и программированием FPGA. Результатом работы такого коллектива будет несколько схем в OrCAD Capture, которые нужно объединить в один проект.

Вот как это можно сделать. Каждый дизайн должен быть оформлен как нижний уровень иерархии. То есть, каждому входу и выходу должен быть поставлен в соответствие иерархический порт. После этого в главном проекте все отдельные схемы собираются в виде иерархических блоков.

Попробуем! Создайте новый проект **Out_Buffer** и разместите его в произвольном месте на диске компьютера.

Теперь откройте проект из предыдущего задания. Вот что должно получиться (**рис. 10-68**):

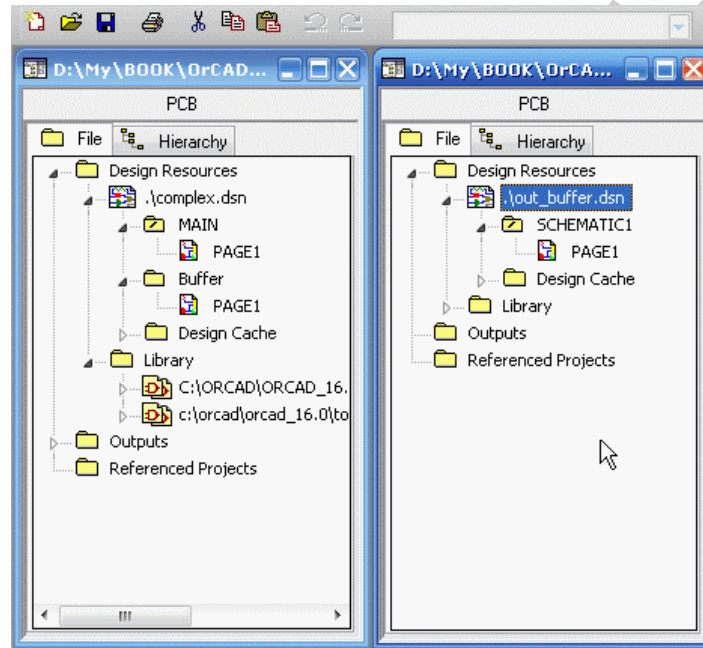


рис. 10-68

С помощью команд **Cut** и **Paste** переместите Schematic **Buffer** в новый проект (**рис. 10-69**):

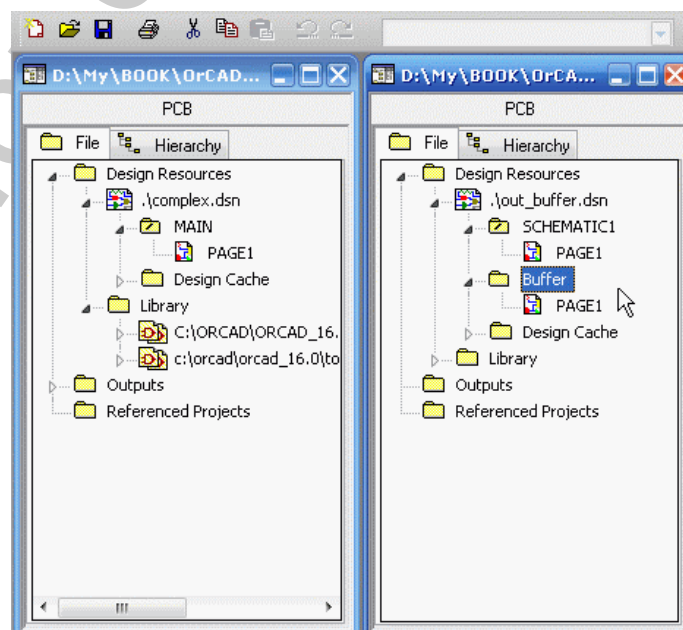


рис. 10-69

Schematic1 из нового проекта, которая была создана автоматически, можно удалить. Я также рекомендую очистить Design Cache в обоих проектах. Это несложно.

Итак, перед нами два проекта (рис. 10-70). Задача: вставить в первый два буфера, схема которых должна быть взята из второго проекта.

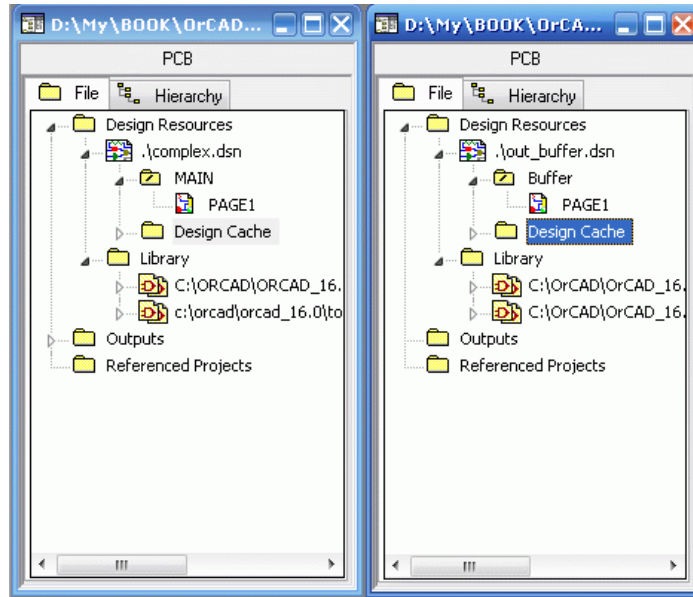


рис. 10-70

Откройте первую схему и удалите из неё оба старых буфера, как будто мы начинаем сначала (рис. 10-71):

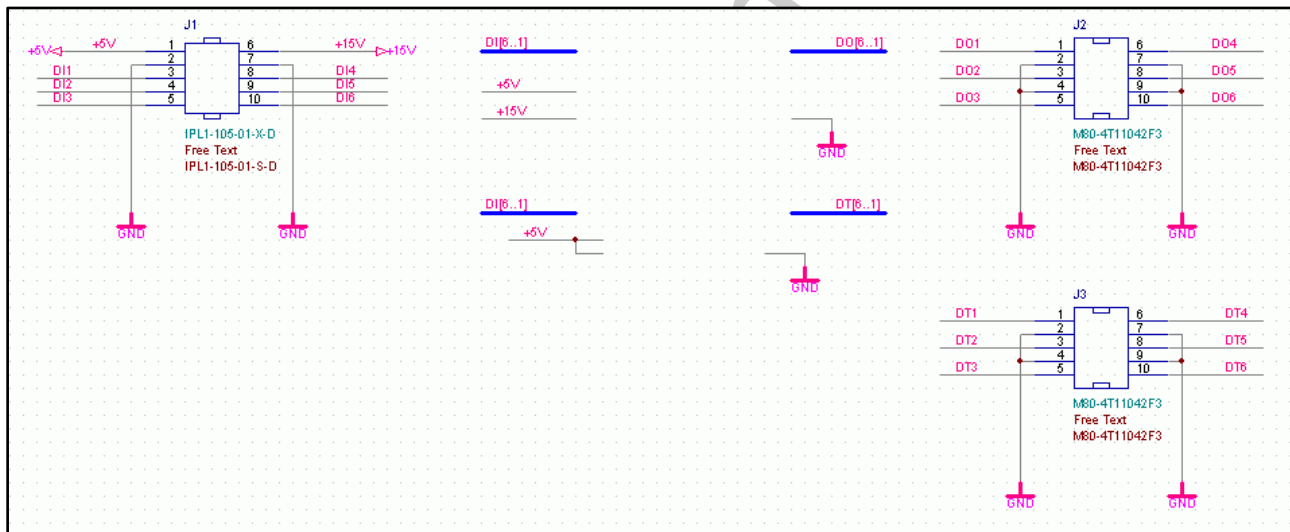


рис. 10-71

Отлично!

Теперь выполним **Place Hierarchical Block**, как мы это уже делали раньше. Заполните необходимые поля (рис. 10-72):

Reference – порядковый номер блока (вместо **H1** можно написать **H?**);

Primitive – No;

Implementation Type – Schematic View;

Path and Filename – укажите путь к дизайну Out_Buffer;

Implementation Name – имя папки Schematic, в которой содержится необходимая схема.

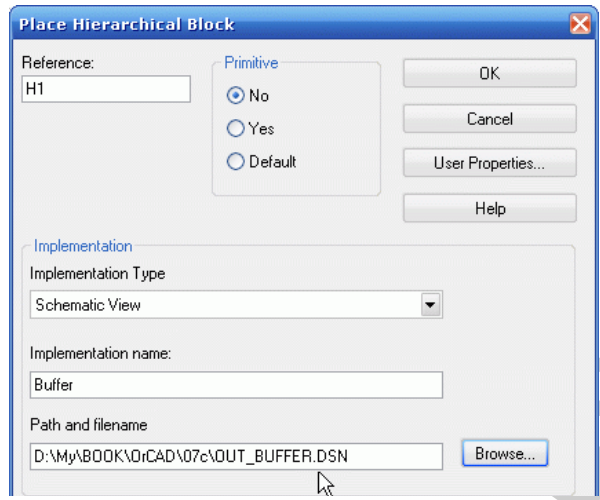


рис. 10-72

Половина дела сделана (рис. 10-73)! Можете убедиться, что по команде **Descent Hierarchy** открывается схема буфера из другого проекта.

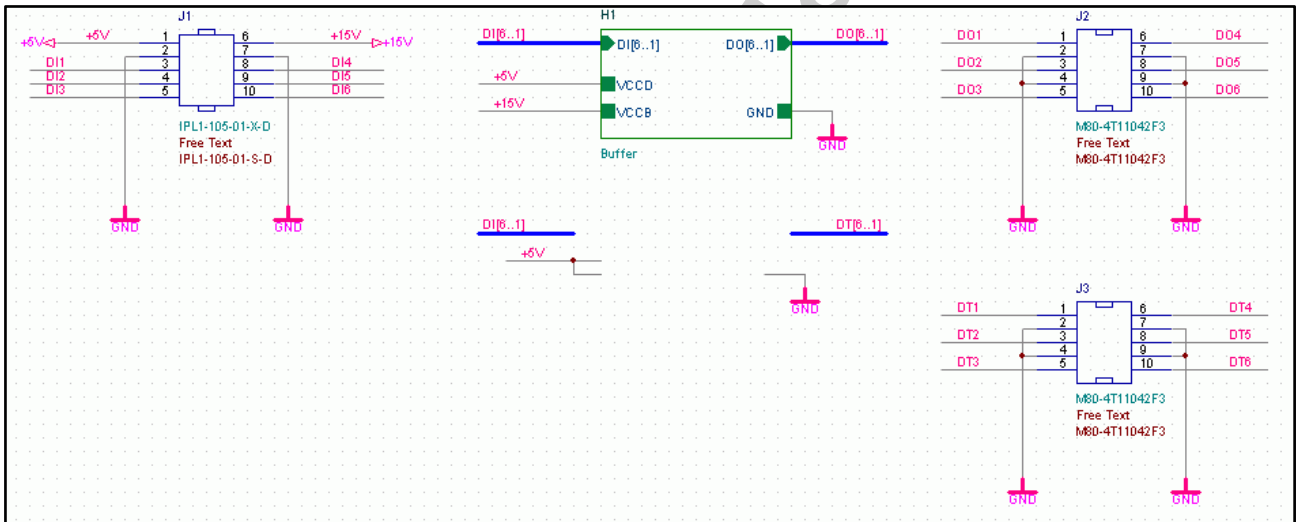


рис. 10-73

Второй буфер можно вставить, просто скопировав первый на новое место.

Вместо этого попробуем вставить элемент **Level Shifter**, который мы создали в предыдущей главе.

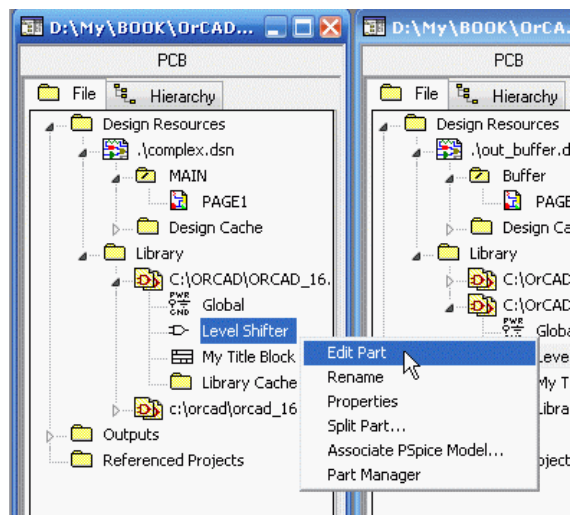


рис. 10-74

Потребуется изменить свойства символа в библиотеке (рис. 10-74).

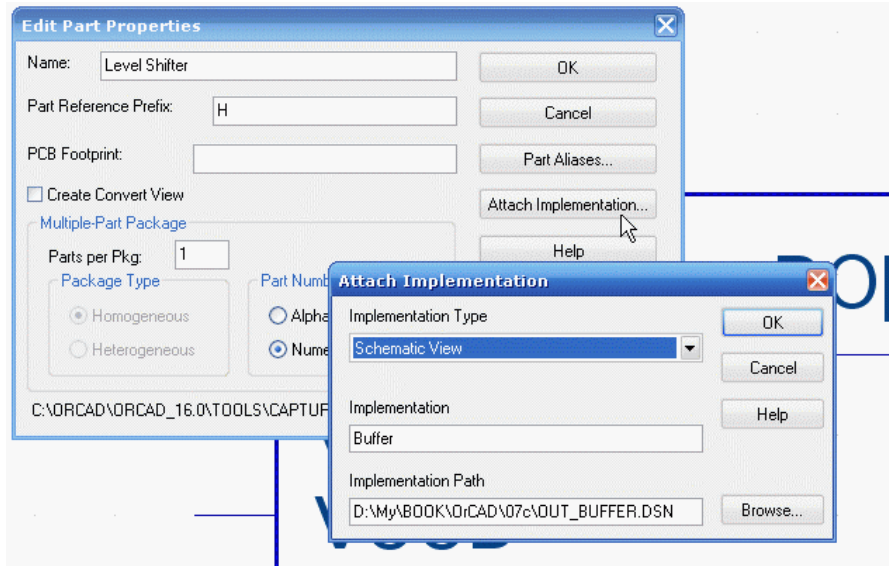


рис. 10-75

Откройте окно **Part Properties**, потом нажмите **<Attach Implementation...>**, затем в поле **Implementation Path** укажите путь к дизайну **Out_Buffer** (рис. 10-75). Сохраните исправленный элемент.

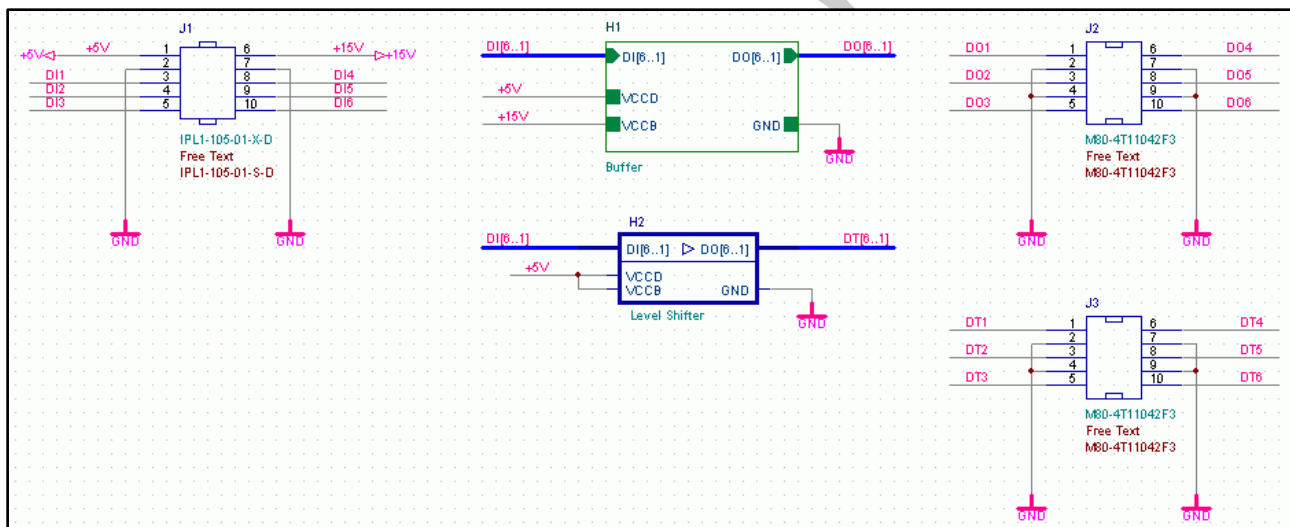


рис. 10-76

Теперь Level Shifter можно вставить в схему (рис. 10-76). Не забудьте отметить его как **Nonprimitive** и выполнить **Annotate**.

Проект готов! Вы можете убедиться в этом, посмотрев дерево иерархии проекта. Уверяю Вас, что и Netlist, и Bill of Materials при необходимости будут созданы без проблем.

В завершение главы хочу заметить, что совсем необязательно было удалять элементы из проекта, редактировать и затем вставлять заново. Достаточно было лишь с помощью **Property Editor** поправить их **Implementation Path**, как показано на рис. 10-77:

Implementation	Implementation Path	Implementation Type	L
Buffer	D:\My\BOOK\OrCAD\07c\OUT_BUFFER.DSN	Schematic View	

рис. 10-77