# REINFORCED CONCRETE STRUCTURE DESIGN ASSISTANT TOOL FOR BEGINNERS

by

Kang-Kyu Choi

A Thesis Presented to the
FACULTY OF THE SCHOOL OF ARCHITECTURE
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements of the Degree
MASTERS OF BUILDING SCIENCE

May 2002

Kang-Kyu Choi G. G. Schierle

## ABSTRACT

## REINFORCED CONCRETE STRUCTURE DESIGN ASSISTANT TOOL FOR BEGINNERS

The objective of this study was a reinforced concrete design tool for architecture students. The tool, a computer program with graphic interface, provides basic concepts for concrete structure calculations and procedures. The graphic interface is expected to help architecture students to understand the design process. The program has four modules: slab, beam, column and footing per American Concrete Institute Code (ACI 318-95).

# DEDICATION

To my parents

# ACKNOWLEDGEMENTS

I would like to thank Professor Goetz Schierle, head of my Thesis Committee, for all his effort, time and patience in helping me to complete this thesis. This thesis would not be possible without his guidance and encouragement.

I would also like to thank the other members of my committee, Professors Dimitry Vergun and Douglas Noble for their time, criticism and suggestion.

Thanks and appreciation are also extended to Professor Marc Shiler. He patiently guided me through the process of making the abstract idea and program and thanks to my classmate and MBS friend whom contributed with value ideas and supporting for my study.

Also, I thank to my friends in KOREA, they always encouraged and trust that I can do what I want to, and I did not feel loneliness because of their cheering via an Internet.

Finally, I would like to take the opportunity to thank my parents and family for having supported me through the all project and studies and Jung-Ran deserves the special appreciation for her support and understanding.

# TABLE OF CONTENTS

Dedication ………………………………………………………………………. ii

Acknowledgements ………………………………………………………….. iii

List of Tables ……………………………………………………………………. vi

List of Figures ………………………………………………………………… vii

Abstract ……………………………………………………………………..…. viii

Hypothesis …………………………………………………………………….. ix

1. Introduction ………………………………………………………………… 1

**Part I: Background** ………………………………………………………….. 3

2. Need for the R.C. Structure Design Program …………………………………. 3
   2.1 Introduction ……………………………………………………...…. 3
   2.2 Review of Existing Programs …………………………………………… 5
      2.2.1 MULTIFRAME 4D ….…………………………………………… 5
      2.2.2 PROKON Calcpad ……………………………………………….. 9

3. Reinforced Concrete Structure ………………………………………………… 12
   3.1 Introduction ………………………………………………………………… 12
   3.2 Building Code Requirement for Structural Concrete (ACI318-95) …….. 14
   3.3 Design Methods of Reinforced Concrete Structure ……………………… 15
      3.3.1 Change of Design Methods according to ACI318 Code …………. 15
      3.3.2 The Working Stress Design (WSD) ……………………………… 16
      3.3.3 The Ultimate Strength Design (USD) …………………………….. 16

4. Review of Structural Calculation on the ACI Code ……………………………. 18
   4.1 Slab …………………………………………………………………………. 18
      4.1.1 Introduction ……………………………………………………….. 18
      4.1.2 Types of Slab ……………………………………………………..…. 20
      4.1.3 Design Procedures ………………………………………………….. 22
   4.2 Beam ………………………………………………………………………….. 24
      4.2.1 Introduction …………………………………………………………. 24
      4.2.2 Types of Beam ………………………………………………………. 25
      4.2.3 Design Procedures ……………….…………………………………..… 26

4.3 Column …………………………………………………………………….. 28
4.3.1 Introduction ………………………………………………………. 28
4.3.2 Types of Column …………………………………………………. 28
4.3.3 Design Procedures ……………………………………...……. 30
4.4 Footing ……………………………………………………………………. 33
4.4.1 Introduction ………………………………………………………. 33
4.4.2 Types of Footing …………………………………………………. 33
4.4.3 Design Procedures ……………………………………………….. 35

**Part II: Reinforced Concrete Structure Design Tool** ………………………… 37

5. Introduction to Reinforced Concrete Structure Designer (RCSD) ……………. 37

6. Slab Design Module ……………………………………………………… 38
6.1 Introduction to Slab Design Module ……………………………………. 38
6.2 One-way Solid Slab Design Module ……………………………………. 38
6.3 Two-way Slab Design Module ………………………………………….. 43
6.4 Flow Chart ………………………………………………………………. 45

7. Beam Design Module ……………………………………………………….. 46
7.1 Introduction to Beam Design Module …………………………………… 46
7.2 Beam Design Module ……………………………………………………. 46
7.3 Flow Chart of Beam Design …………………..……………………… 49
7.4 Flow Chart of Shear Check ……………………………………...….. 50
7.5 Flow Chart of Deflection Check ………………………………..….. 51

8. Column Design Module …………………………………………………….. 52
8.1 Introduction to Column Design Module …………………………………. 52
8.2 Column Analysis Program ………………………………………………. 53
8.3 Simplified P-M Interaction Diagram ……………………………………. 54
8.4 General Equation for Simplified P-M Interaction Diagram …………….. 56
8.5 Column Design Module …………………………………………………. 60
8.6 Flow Chart ………………………………………………………………. 62

9. Footing Design Module …………………………………………………..…. 63
9.1 Introduction to Footing Design Module …………………………………. 63
9.2 Flow Chart of Individual Column Footing ………………………………. 65
9.3 Flow Chart of Individual Column Footing ………………………………. 66

10. Conclusions ……………………………………………………………………. 67

**III Bibliography** ………………………………………………………………. 69

**IV Appendix** ……..…………………………………………………………. 71

**List of Tables**

Table 2-1: The Percentage of Structure Classes …………………………………… 3
Table 3-1: Factored load combinations for determining required strength U …….. 17
Table 3-2: Strength reduction factors in the ACI Code …………………………… 17
Table 8-1: Simplified P-M interaction equation ………………………..……… 56

**List of Figures**

Fig. 2-1: Multiframe 4D Main window …………………………………………… 6
Fig. 2-2: Multiframe 4D Input window …………………………………………… 6
Fig. 2-3: Multiframe 4D Section Maker window ………………………………… 7
Fig. 2-4: Multiframe 4D Moment analysis window ……………………………..….. 7
Fig. 2-5: Multiframe 4D Deflection analysis window …………………………..… 8
Fig. 2-6: Multiframe 4D Detail member analysis window ……………………..… 8
Fig. 2-7: PROKON Calcpad Main page ………………………………………….. 9
Fig. 2-8: PROKON Calcpad Rectangular column input page ………………..….. 10
Fig. 2-9: PROKON Calcpad Rectangular column design page…………………… 10
Fig. 2-10: PROKON Calcpad Rectangular column calculation sheet page ………. 11
Fig. 2-11: PROKON Calcpad Rectangular column reinforcement data page ……. 11
Fig. 4-1: One-way slab design concept …………………………………………… 18
Fig. 4-2: Typical type of slabs ……………………………………………………. 21
Fig. 4-3: Reinforced rectangular beam …………………………………………… 24
Fig. 4-4: Common shapes for concrete beam …………………………………….. 25
Fig. 4-5: Column types …………………………………………………………… 29
Fig. 4-6: The column types depending on applied load ………………………….. 30
Fig. 4-7: Eccentrically loaded columns ………………………………………….. 30
Fig. 4-8: Footing types …………………………………………………………… 35
Fig. 6-1: Superimposed Dead Load Calculator ………………………………….. 39
Fig. 6-2: MOMENT tab of One-way slab module ……………………………….. 40
Fig. 6-3: REINFORCEMENT tab of One-way slab module …………………….. 41
Fig. 6-4: Shear and deflection checks of One-way slab module ………………… 42
Fig. 6-5: Two-way slab minimum thickness output ……………………………… 43
Fig. 6-6: Two-way slab deflection check tab ……………………………………..… 44
Fig. 7-1: Beam tributary area and support type input …………………………… 46
Fig. 7-2: Design the reinforcement of beam module …………………………….. 47
Fig. 7-3: The shear check and stirrup design …………………………………….. 48
Fig. 8-1: Column interaction diagram …………………………………………… 52
Fig. 8-2: Simplified Column Analysis Program ………………………………… 53
Fig. 8-3: SCAP simplified P-M Interaction diagram …………………………….. 54
Fig. 8-4: P-M Interaction diagram with 15% cover thickness ………………..….. 55
Fig. 8-5: Simplified P-M Interaction diagram …………………………………… 57
Fig. 8-6: Axial stress and steel strength ………………………………………….. 58
Fig. 8-7: Axial stress and concrete strength ……………………………………… 58
Fig. 8-8: Strength reduction in % due to moment in column ……………………. 59
Fig. 8-9: CHECK tab of column design module ………………………………… 61
Fig. 9-1: The possible footing size calculation table and drawing ……………….. 63
Fig. 9-2: REINFORCEMENT tab of individual column footing ………………… 64
Fig. 9-3: The result of footing design and 3D rendered images ………………..… 64

## Abstract:

The objective of this study was a reinforced concrete design tool for architecture students. The tool, a computer program with graphic interface, provides basic concepts for concrete structure calculations and procedures. The graphic interface is expected to help architecture students to understand the design process. The program has four modules: slab, beam, column and footing per American Concrete Institute Code (ACI 318-95).

Key Words: Concrete design, Concrete structures, ACI Code, Concrete design teaching tool, RC Concrete software.

## Hypothesis

This simplified reinforced concrete structure design program for architecture students, based on the American Concrete Institute Code (ACI 318-95), is expected to help architecture students to design sound concrete structures.

# 1. Introduction

Reinforced concrete structures are one of the most popular structure systems. Many architecture students are using reinforced concrete structure systems for their designs. But there are many cases where they design structurally questionable buildings because they are trying to express their design ideas with limited knowledge about R.C. Design. Frequently the structural member design would not be their primary focus. Although there is the possibility that excessive structural considerations may disturbing their search for unique designs, basic structural calculation is important for design. Structurally sound solutions can make their design concepts closer to reality.

Unfortunately most architecture schools concentrate their curriculum on visual design education rather than a balanced education of design and structure. The balanced education does not mean equal class time for structural and design classes. But it is essential that students can at least discriminate that their design has a reasonable structure. Many students use the commonly available books on architectural graphic standards as a reference. But they are not applicable to many different conditions. Furthermore, reinforced concrete structures need a lot of calculations and different condition inputs because it is a composite material of concrete and steel.

The ***Reinforced Concrete Structure Design*** program (RCSD), which has been developed for this thesis, can help architecture students and users to analyze their designs and understand structural fundamentals. Although there are many reinforced concrete structure programs, most programs are targeting advanced level users who have a background in structural engineering.

The RCSD program is for beginner level users such as architecture undergraduate and graduate students with limited knowledge about structures. For this, it provides a graphical input method and a step-by-step calculation procedure to help users. With this program, it is possible for the user to design basic structural parts such as slab, beam, column and footing. Also the program is based on the American Concrete Institute Code. The ultimate goal of this program is that users can analyse their own designs using this program and determine structural proportions of their design idea.

# Part I: BACKGROUND STUDY

## 2. Need for the Reinforced Concrete (RC) Structure Design Program

### *2.1 Structural Education of Architecture Students*

Many architecture schools do not teach architectural engineering but only architecture and the schools that have architectural engineering usually are part of an engineering school rather than an architecture department. Most architecture schools provide only a few structure classes for students, not enough to fully understand structural design use in their project. Even the schools that are ranked as The Best Architecture Graduate Schools (U.S.News & World Report Inc, 2001) offer less than 15% of structure related classes in their curriculums (Table 2-1).

| | Total Classes | Structure Classes | % |
|---|---|---|---|
| M.I.T. | 151 | 15 | 9.9 |
| Princeton University | 32 | 3 | 9.3 |
| Columbia University | 67 | 8 | 11.9 |
| Yale University | 31 | 3 | 9.6 |
| University of California Berkeley | 105 | 10 | 9.5 |
| University of Virginia | 12 | 2 | 16.6 |
| University of Pennsylvania | 13 | 1 | 7.6 |
| Georgia Institute of Technology | 47 | 7 | 14.9 |
| Total | 458 | 49 | 10.9 |

Table 2-1: The Percentage of Structure Classes

Therefore architecture students do not have enough opportunities to study structural education even though they may want to study structure systems in relation to architecture.

Also it is hard to say that the best solution would be that architecture schools increase structure classes because it is almost impossible to teach detailed structural calculation methods to architecture students like is done in engineering schools. Architecture students do not need to know the complete details of structural systems but rather the intuitive information about structural safety of their own designs. Reinforced concrete calculations are particularly repetitive calculations. This is one reason why a reinforced concrete structures design program is valuable for architecture students. Since architecture students use scale models or use computer graphic modeling to understand and present their design, by using a computer analysis program users can save time and complement their lack of structural knowledge with the presented program.

## *2.2 Review of Existing Structural Programs*

The two existing sofrware programs are inapprorpriate for use by architecture students as described earlier.

### 2.2.1 MULTIFRAME 4D(Daystar Software, Inc.)

Multiframe is a 2D and 3D self-executable static and dynamic analysis program from Daystar Software Inc. This software provides a good graphical interface and comprehensive analysis capabilities. Multiframe can analyze not only reinforced concrete structures but also all types of framed structures.

It has its own library of common structural material and the user can analyze steel, concrete or timber frames using this material library. However, reinforced concrete structures have different material section properties depending on concrete, steel bar strength and ratio of reinforcement. Usually the user has to make a new material library to analyze a reinforced concrete structure (Fig. 2-3).

Multiframe is developed for experienced users. The user can input their building manually or import AutoCAD files.

Multiframe provides a lot of output data, such as stresses of each member, moments and even animated deflections (Fig. 2-4).

The program is a bit complex for beginners. It will not analyze a structure without flawless data input including section properties, condition of joint type and load. The user has to input all data, which is hard for beginners. For architecture students, a design program is more useful than an analysis program.

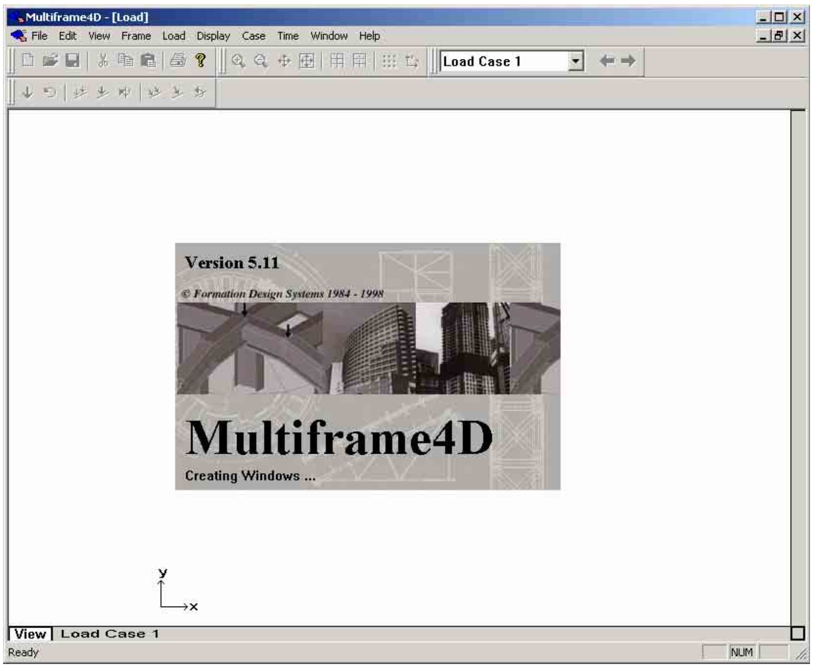



Fig. 2-1: Multiframe 4D Main window

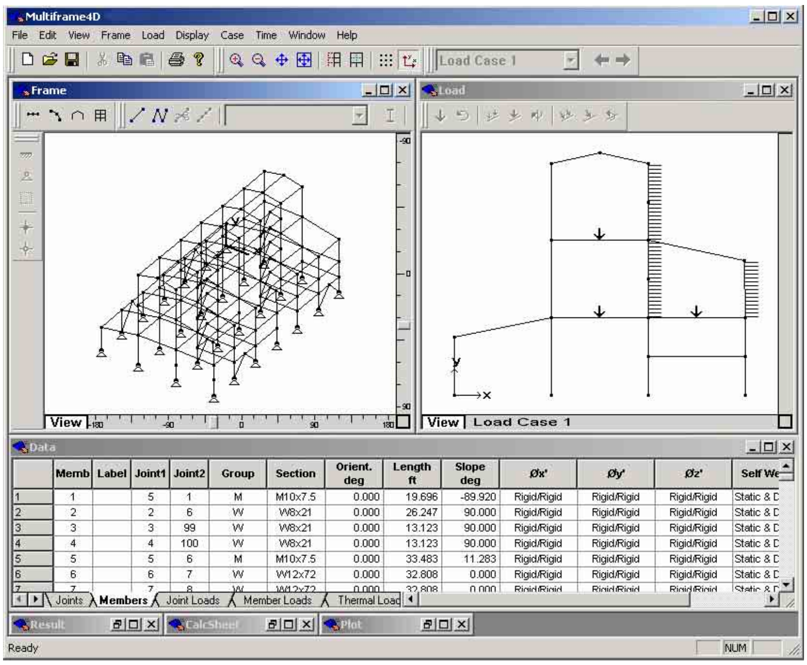



Fig. 2-2: Multiframe 4D Input window

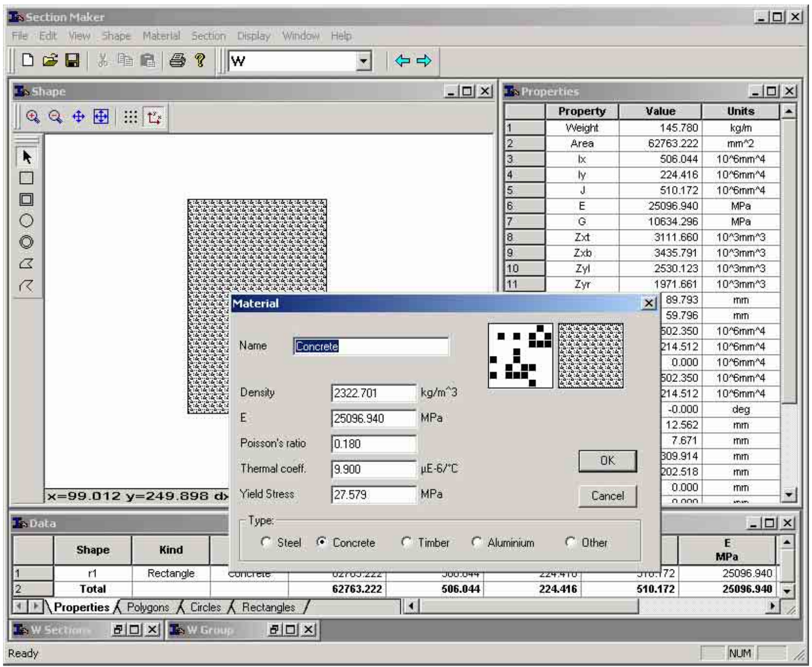



Fig. 2-3: Multiframe 4D Section Maker window

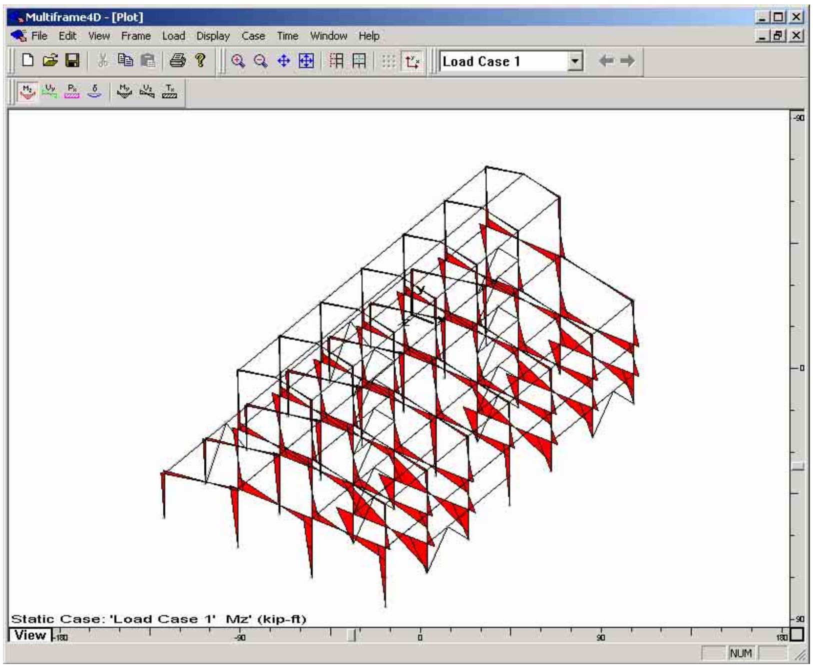



Fig. 2-4: Multiframe 4D Moment analysis window

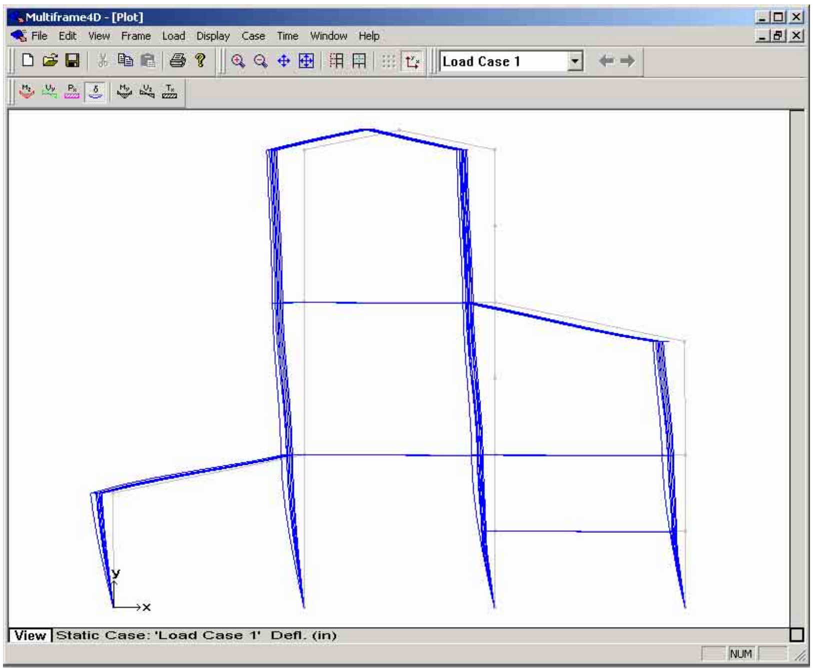



Fig. 2-5: Multiframe 4D Deflection analysis window

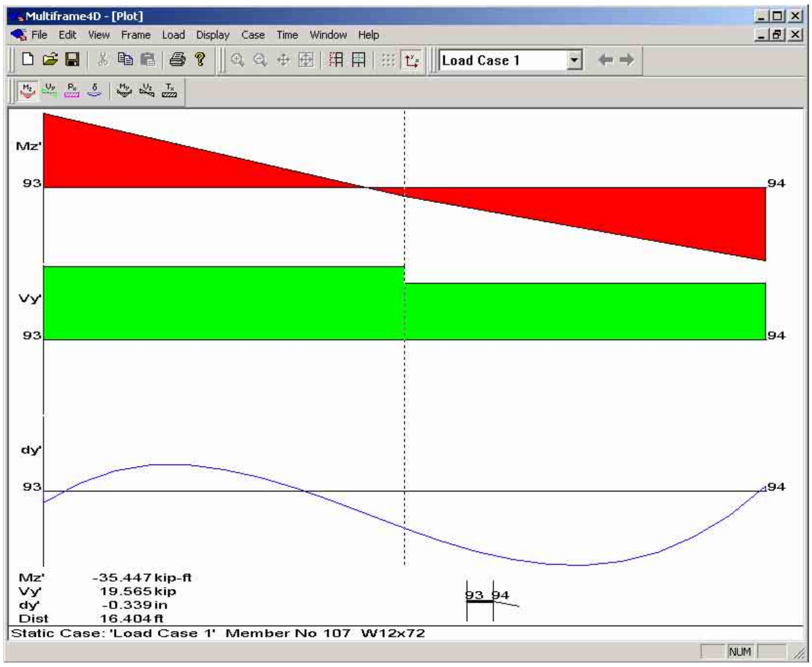



Fig. 2-6: Multiframe 4D Detail member analysis window

### 2.2.2 PROKON Calcpad (Prokon Software Consultant Ltd.)

The PROKON Calcpad is a structural analysis and design software for concrete, steel and timber design. This program has been developed for average level users. It provides a graphical user interface, continuous error checking during the input phase and table editor. This is really helpful to find and fix input problems.

The PROKON Calcpad cannot analyze a whole structure like Multiframe because it has discrete calculation modules, such as concrete slab, rectangular column, retaining wall and footing. This modular program is easy to understand but the users has to calculate factored load for each part because the program cannot calculate the load and tributary area without overall building conditions.

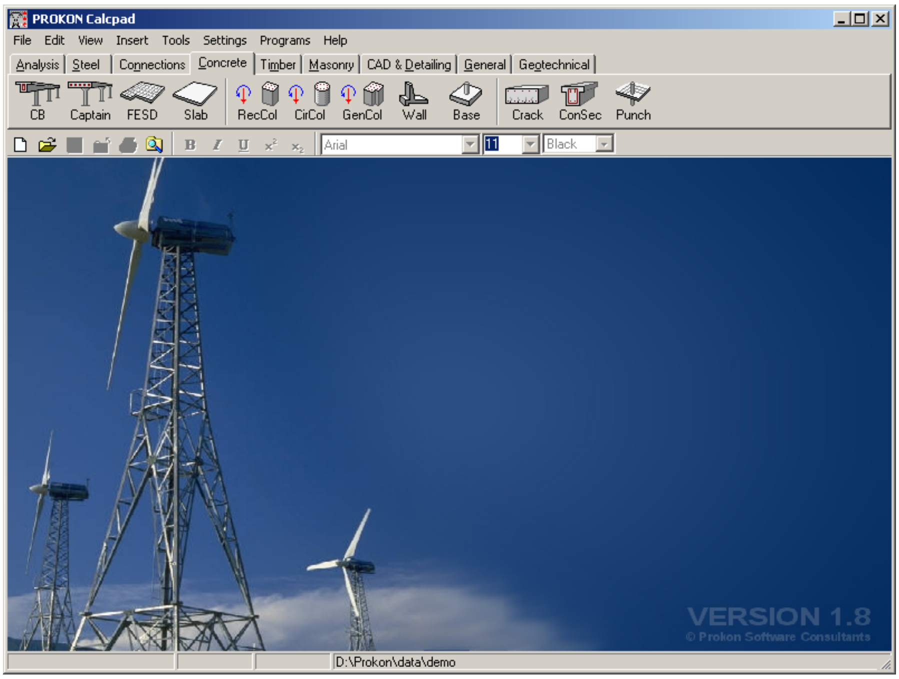


Fig. 2-7: PROKON Calcpad Main page

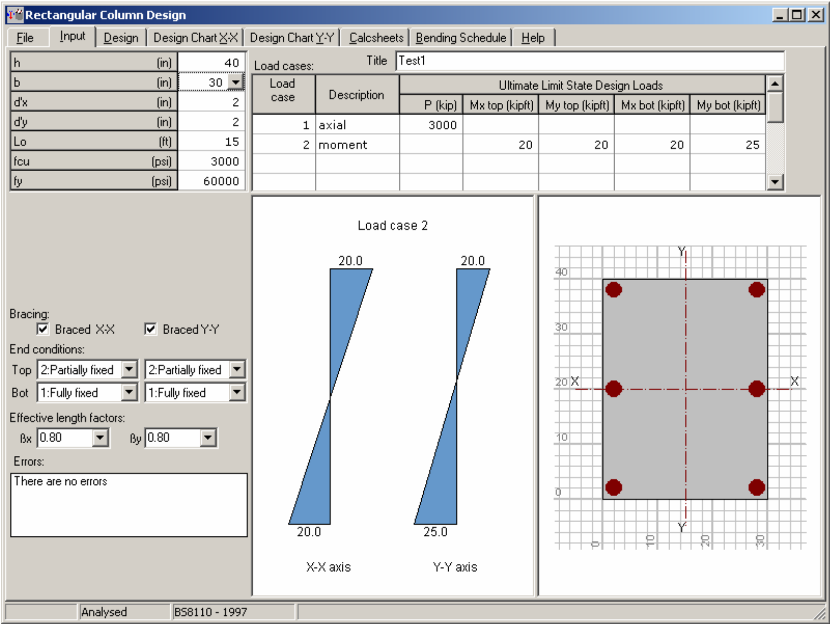



Fig. 2-8: PROKON Calcpad Rectangular column input page

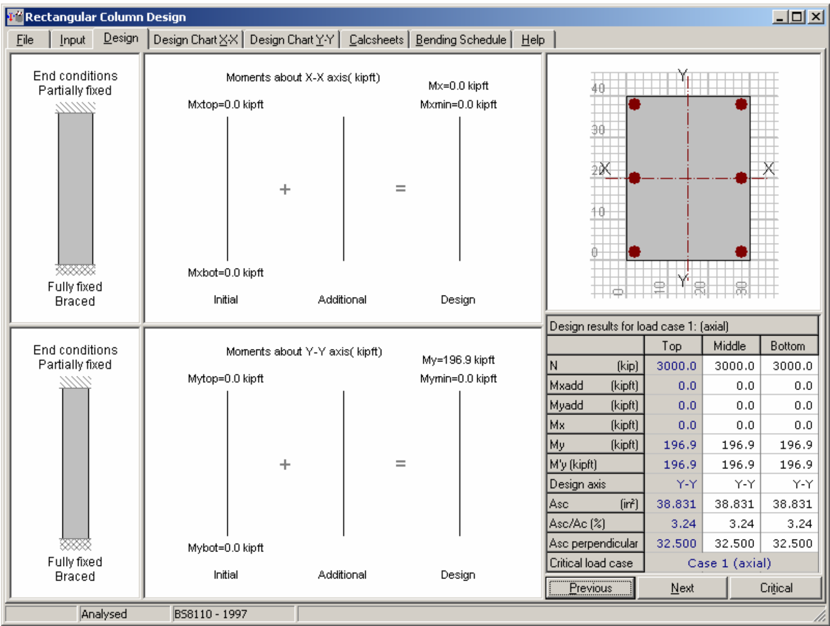



Fig. 2-9: PROKON Calcpad Rectangular column design page

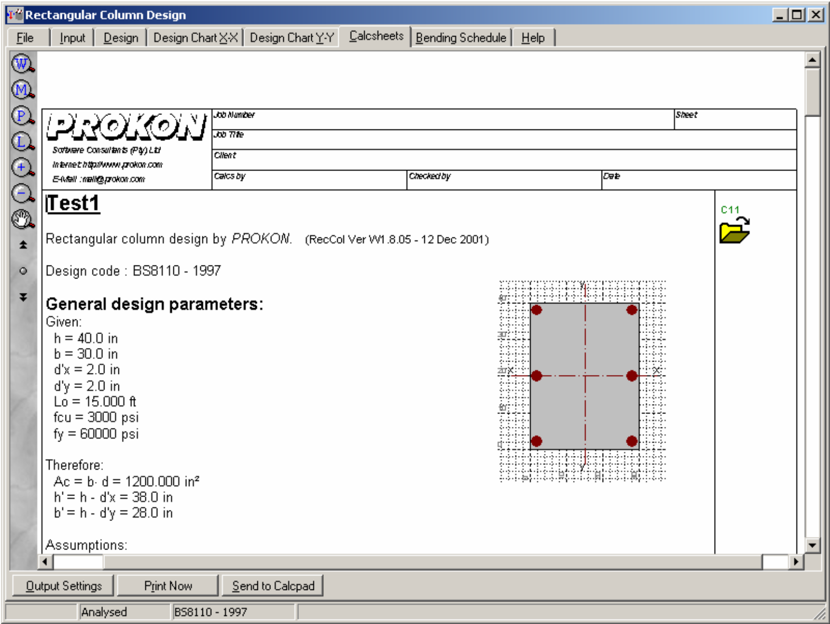



Fig. 2-10: PROKON Calcpad Rectangular column calculation sheet page

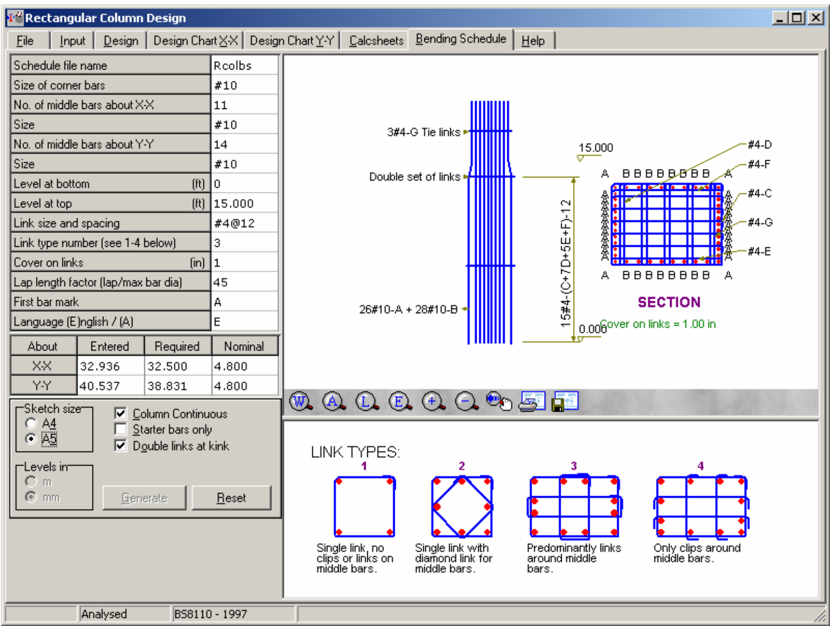



Fig. 2-11: PROKON Calcpad Rectangular column reinforcement data page

# 3. Reinforced Concrete Structure

## *3.1 Introduction*

Concrete is one of the most popular materials for buildings because it has high compressive strength, flexibility in its form and it is widely available. The history of concrete usage dates back for over a thousand years. Contemporary cement concrete has been used since the early nineteenth century with the development of Portland cement.

Despite the high compressive strength, concrete has limited tensile strength, only about ten percent of its compressive strength and zero strength after cracks develop.

In the late nineteenth century, reinforcing materials, such as iron or steel rods, began to be used to increase the tensile strength of concrete. Today steel bars are used as common reinforcing material. Usually steel bars have over 100 times the tensile strength of concrete; but the cost is higher than concrete. Therefore, it is most economical that concrete resists compression and steel provides tensile strength. Also it is essential that concrete and steel deform together and deformed reinforcing bars are being used to increase the capacity to resist bond stresses.

Advantages of reinforced concrete can be summarized as follows (Hassoun, 1998).

1. It has a relatively high compressive strength.
2. It has better resistance to fire than steel or wood
3. It has a long service life with low maintenance cost.

4. In some types of structures, such as dams, piers, and footing, it is the most economical structural material.
5. It can be cast to take any shape required, making it widely used in precast structural components.

Also, disadvantages of reinforced concrete can be summarized as follows:

1. It has a low tensile strength (zero strength after cracks develop).
2. It needs mixing, casting, and curing, all of which affect the final strength of concrete.
3. The cost of the forms used to cast concrete is relatively high. The cost of form material and artisanry may equal the cost of concrete placed in the forms.
4. It has a lower compressive strength than steel (about 1/10, depending on material), which requires large sections in columns of multistory buildings.
5. Cracks develop in concrete due to shrinkage and the application of live loads.

### *3.2 Building Code Requirement for Structural Concrete (ACI318-95)*

Many countries have building codes to define material properties, quality controls, minimum size, etc for safety constructions. However, the United States does not have an official government code. However, the Uniform Building Code (UBC) and other model codes are adapted by jurisdictions, such as Cities, or States as governing codes. Material and methods are tested by private or public organizations. They develop, share, and disseminate their result and knowledge for adoption by jurisdictions. The American Concrete Institute (ACI) is leading the development of concrete technology. The ACI has published many references and journals. *Building Code Requirement for Structural Concrete (ACI318 Code)* is a widely recognized reinforced concrete design and construction guide. Although the ACI Code dose not have official power of enforcement, it is generally adapted as authorized code by jurisdictions not only in United States but also many countries. The ACI318 Code provides the design and construction guide of reinforced concrete. ACI has been providing new codes depending on the change of design methods and strength requirement.

### *3.3 Design Methods of Reinforced Concrete Structure*

Two major calculating methods of reinforced concrete have been used from early 1900's to current. The first method is called Working Stress Design (WSD) and the second is called Ultimate Strength Design (USD). Working Stress Design was used as the principal method from early 1900's until the early 1960's. Since Ultimate Strength Design method was officially recognized and permitted from ACI 318-56, the main design method of ACI 318 Code has gradually changed from WSD to USD method. The program of this thesis is based on ACI 318-95 Code USD Method, published in 1995.

#### **3.3.1 Change of Design Methods according to ACI 318 Code** (PCA, 1999).

| | |
|---|---|
| ACI 318-56: | USD was first introduced (1956) |
| ACI 318-63: | WSD and USD were treated on equal basis. |
| ACI 318-71: | Based entirely on strength Method (USD) |
| | WSD was called Alternate Design Method (ADM). |
| ACI 318-77: | ADM relegated to Appendix B |
| ACI 318-89: | ADM back to Appendix A |
| ACI 318-95: | ADM still in Appendix A |
| | Unified Design Provision was introduced in Appendix B |
| ACI 318-02: | ADM was deleted from Appendix A (ACI,2002) |

#### **3.3.2 The Working Stress Design (WSD)**

Traditionally, elastic behavior was used as basis for the design method of reinforced concrete structures. This method is known as Working Stress Design (WSD) and also called the Alternate Design Method or the Elastic Design Method. This design concept is based on the elastic theory that assumes a straight-line stress distribution along the depth of the concrete section.

To analyze and design reinforced concrete members, the actual load under working conditions, also called service load condition, is used and allowable stresses are decided depending on the safety factor. For example allowable compressive bending stress is calculated as $0.45f'_{c}$. If the actual stresses do not exceed the allowable stresses, the structures are considered to be adequate for strength.

The WSD method is easier to explain and use than other method but this method is being replaced by the Ultimate Strength Design method. ACI 318 Code treats the WSD method just in a small part.

### 3.3.3 The Ultimate Strength Design (USD)

The Ultimate Strength Design method, also called Strength Design Method (SDM), is based on the ultimate strength, when the design member would fail. The USD method provides safety not by allowable stresses as for the ASD method but by factored loads, nominal strength and strength reduction factors θ, both defined by the ACI code.

The load factors are 1.7 for live load and 1.4 for dead load. Other factors are given in Table 3-1.

| Condition | Factored load or load effect U |
|---|---|
| Basic | U = 1.4D + 1.7L |
| Winds | U = 0.75(1.4D + 1.7L + 1.7W)<br>U = 0.9D + 1.3W<br>U = 1.4D + 1.7L |
| Earthquake | U = 0.75(1.4D + 1.7L + 1.87E)<br>U = 0.9D + 1.43E<br>U = 1.4D + 1.7L |
| Earth pressure | U = 1.4D + 1.7L + 1.7H<br>U = 0.9D + 1.7H<br>U = 1.4D + 1.7L |
| Settlement, creep, shrinkage, or temperature change effects | U = 0.75(1.4D + 1.4T + 1.7L)<br>U = 1.4(D + T) |

Table 3-1: Factored load combinations for determining required strength U

However, deflections are based on service load rather than factored load. The strength reduction factors are given in Table 3-2. Different factors are used for beams, tied column, or spiral column.

| Kind of strength | Strength reduction factor $\phi$ |
|---|---|
| Flexure, without axial load | 0.90 |
| Axial tension<br>Axial compression with flexure | 0.90 |
| Axial compression<br>Axial compression with flexure member | 0.70 |
| Axial compression<br>Axial compression with flexure member with spiral reinforcement | 0.75 |
| Shear and torsion | 0.85 |
| Bearing on concrete | 0.70 |

Table 3-2: Strength reduction factors in the ACI Code (Nilson, 1997)

## 4. Review of Structural Calculation on the ACI Code

## 4.1 Slab

### 4.1.1 Introduction

The slab provides a horizontal surface and is usually supported by columns, beams or walls. Slabs can be categorized into two main types: one-way slabs and two-way slabs.

One-way slab is the most basic and common type of slab. One-way slabs are supported by two opposite sides and bending occurs in one direction only. Two-way slabs are supported on four sides and bending occurs in two directions. One-way slabs are designed as rectangular beams placed side by side (Fig. 4-1).

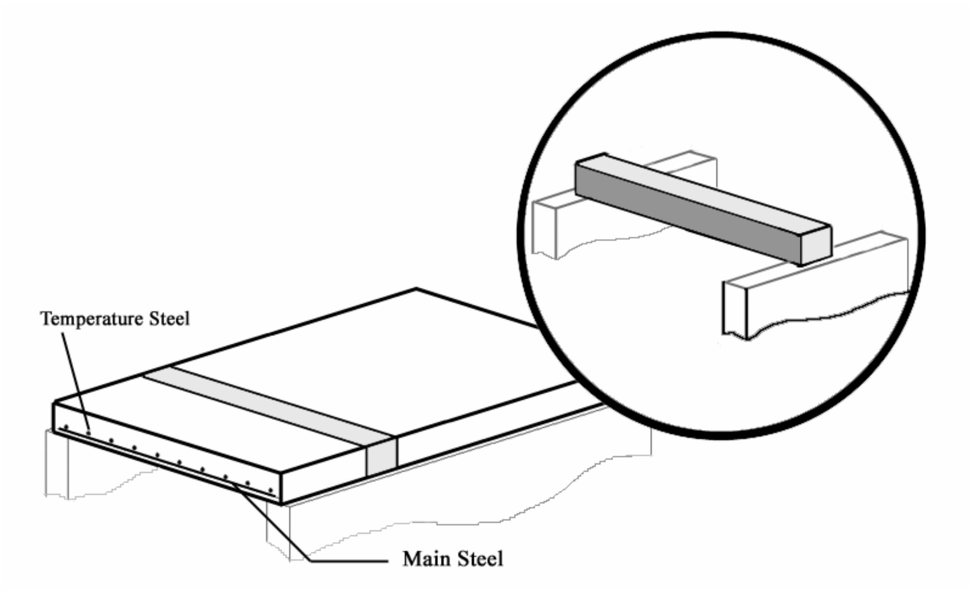



Fig. 4-1: One-way slab design concept

However, slabs supported by four sides may be assumed as one-way slab when the ratio of lengths to width of two perpendicular sides exceeds 2. Although while such slabs transfer their loading in four directions, nearly all load is transferred in the short direction.

Two-way slabs carry the load to two directions, and the bending moment in each direction is less than the bending moment of one-way slabs. Also two-way slabs have less deflection than one-way slabs. Compared to one-way slabs, Calculation of two-way slabs is more complex. Methods for two-way slab design include Direct Design Method (DDM), Equivalent frame method (EFM), Finite element approach, and Yield line theory. However, the ACI Code specifies two simplified methods, DDM and EFM.

### 4.1.2 Types of Slabs

- One-way slabs

1. One-way Beam and slab / One-way flat slab:

   These slabs are supported on two opposite sides and all bending moment and deflections are resisted in the short direction. A slab supported on four sides with length to width ratio greater than two, should be designed as one-way slab.

2. One-way joist floor system:

   This type of slab, also called ribbed slab, is supported by reinforced concrete ribs or joists. The ribs are usually tapered and uniformly spaced and supported on girders that rest on columns.

- Two-way slab

  1. Two-way beam and slab:

     If the slab is supported by beams on all four sides, the loads are transferred to all four beams, assuming rebar in both directions.

  2. Two-way flat slab:

     A flat slab usually does not have beams or girders but is supported by drop panels or column capitals directly. All loads are transferred to the supporting column, with punching shear resisted by drop panels.

  3. Two-way waffle slab:

     This type of slab consists of a floor slab with a length-to-width ratio less than 2, supported by waffles in two directions.

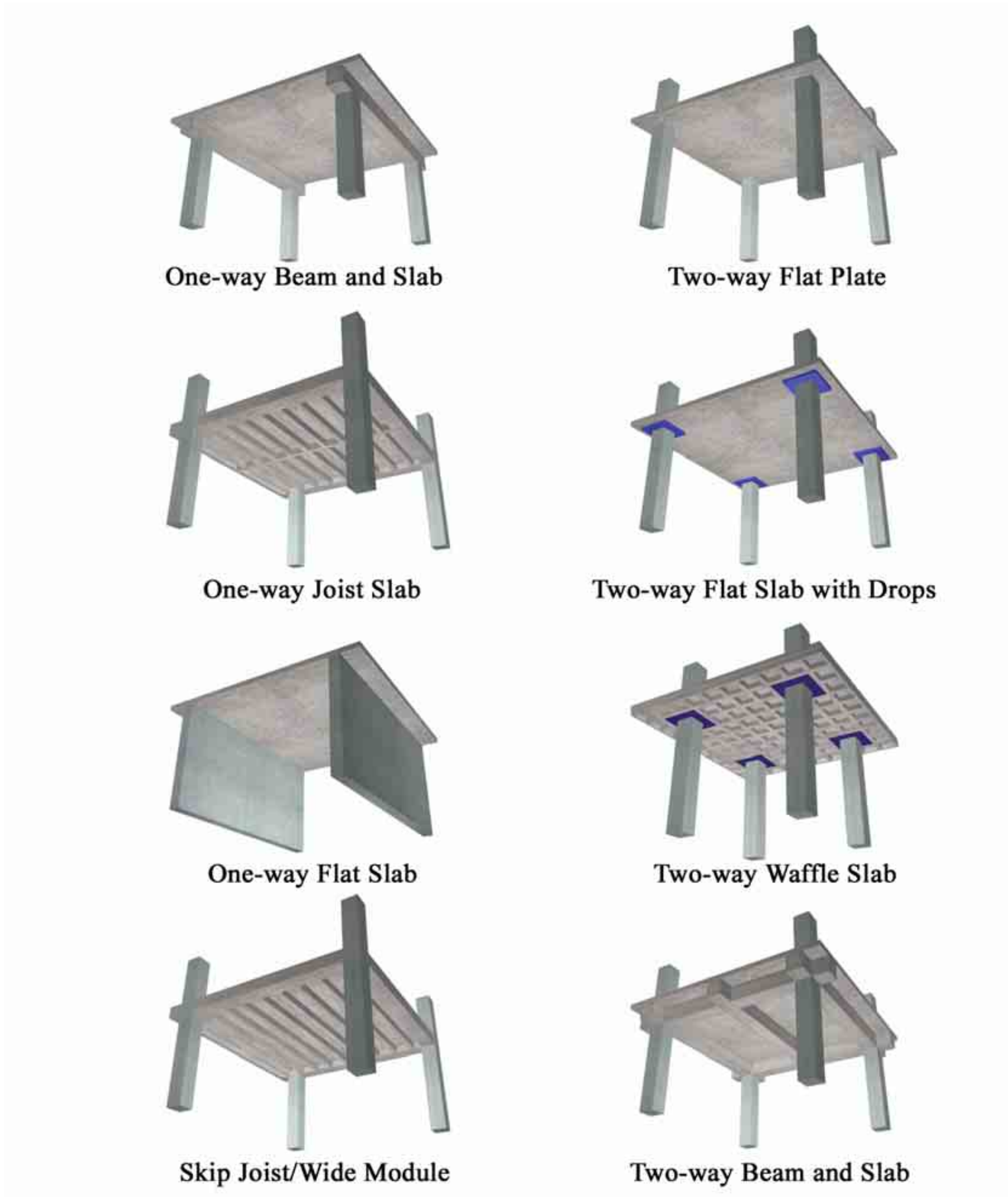



Fig. 4-2: Typical type of slabs (ACI,1994)

### 4.1.3 Design Procedure

- One-way slab design
    1. Decide the type of slab according to aspect ratio of long and short side lengths.
    2. Compute the minimum thickness based on ACI Code.
    3. Compute the slab self-weight and total design load.
    4. Compute factored loads (1.4 DL + 1.7 LL).
    5. Compute the design moment.
    6. Assume the effective slab depth.
    7. Check the shear.
    8. Find or compute the required steel ratio.
    9. Compute the required steel area.
    10. Design the reinforcement (main and temperature steel).
    11. Check the deflection.

- Two-way slab design procedure by the Direct Design Method
    1. Decide the type of slab according to aspect ratio of long and short side lengths.
    2. Check the limitation to use the DDM in ACI Code. If limitations are not met, the DDM can not be used.
    3. Determine and assume the thickness of slab to control deflection.
    4. Compute the slab self-weight and total design load.
    5. Compute factored loads (1.4 DL + 1.7 LL).

6. Check the slab thickness against one-way shear and two-way shear.
7. Compute the design moment.
8. Determine the distribution factor for the positive and negative moments using ACI Code.
9. Determine the steel reinforcement of the column and middle strips.
10. Compute the unbalanced moment and check if it is adequate.

### *4.2 Beam*

### 4.2.1 Introduction

Beams can be described as members that are mainly subjected to flexure and it is essential to focus on the analysis of bending moment, shear, and deflection. When the bending moment acts on the beam, bending strain is produced. The resisting moment is developed by internal stresses. Under positive moment, compressive strains are produced in the top of beam and tensile strains in the bottom. Concrete is a poor material for tensile strength and it is not suitable for flexure member by itself. The tension side of the beam would fail before compression side failure when beam is subjected a bending moment without the reinforcement. For this reason, steel reinforcement is placed on the tension side. The steel reinforcement resists all tensile bending stress because tensile strength of concrete is zero when cracks develop. In the Ultimate Strength Design (USD), a rectangular stress block is assumed (Fig. 4-3).

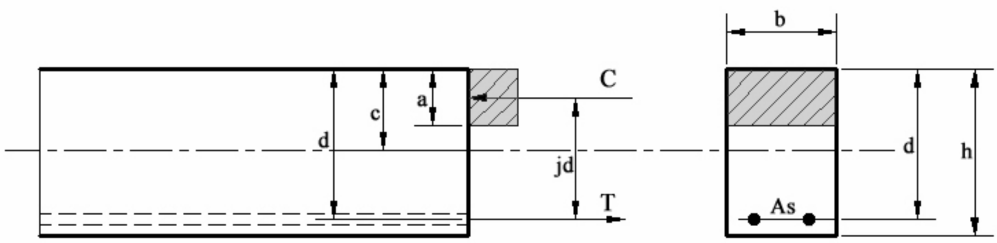
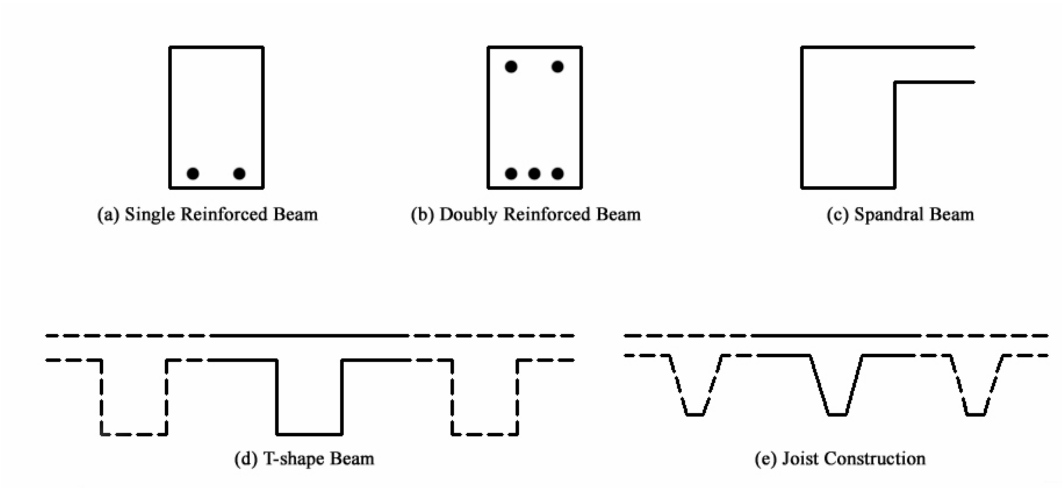



Fig 4-3: Reinforced rectangular beam (Ambrose, 1997)

As shown Fig. 4-3, the dimensions of the compression force is the product of beam width, depth and length of compressive stress block. The design of beam is initiated by the calculation of moment strengths controlled by concrete and steel.

### 4.2.2 Types of Beam

Fig. 4-4 shows the most common shapes of concrete beams: single reinforced rectangular beams, doubly reinforced rectangular beams, T-shape beams, spandrel




beams, and joists.

Fig. 4-4: Common shapes of concrete beam (Spiegel, 1998)

In cast–in-place construction, the single reinforced rectangular beam is uncommon. The T-shape and L-shape beams are typical types of beam because the beams are built monolithically with the slab. When slab and beams are poured together, the slab on the beam serves as the flange of a T-beam and the supporting beam below slab is the stem or web. For positive applied bending moment, the bottom of section produces the tension and the slab acts as compression flange. But negative bending on a rectangular beam puts the stem in compression and the flange is ineffective in tension. Joists consist of spaced ribs and a top flange.

### 4.2.3 Design Procedure

- Rectangular Beam
    1. Assume the depth of beam using the ACI Code reference, minimum thickness unless consideration the deflection.
    2. Assume beam width (ratio of with and depth is about 1:2).
    3. Compute self-weight of beam and design load.
    4. Compute factored load (1.4 DL + 1.7 LL).
    5. Compute design moment ($M_u$).
    6. Compute maximum possible nominal moment for singly reinforced beam ($\phi M_n$).
    7. Decide reinforcement type by Comparing the design moment ($M_u$) and the maximum possible moment for singly reinforced beam ($\phi M_n$). If $\phi M_n$ is less than $M_{u,}$ the beam is designed as a doubly reinforced beam else the beam can be designed with tension steel only.
    8. Determine the moment capacity of the singly reinforced section. (concrete-steel couple)
    9. Compute the required steel area for the singly reinforced section.

10. Find necessary residual moment, subtracting the total design moment and the moment capacity of singly reinforced section.
11. Compute the additional steel area from necessary residual moment.
12. Compute total tension and compressive steel area.
13. Design the reinforcement by selecting the steel.
14. Check the actual beam depth and assumed beam depth.

- T-shape Beam
    1. Compute the design moment ($M_u$).
    2. Assume the effective depth.
    3. Decide the effective flange width (*b*) based on ACI criteria.
    4. Compute the practical moment strength ($\phi M_n$) assuming the total effective flange is supporting the compression.
    5. If the practical moment strength ($\phi M_n$) is bigger than the design moment ($M_u$), the beam will be calculated as a rectangular T-beam with the effective flange width *b*. If the practical moment strength ($\phi M_n$) is smaller than the design moment ($M_u$), the beam will behave as a true T-shape beam.
    6. Find the approximate lever arm distance for the internal couple.
    7. Compute the approximate required steel area.
    8. Design the reinforcement.
    9. Check the beam width.
    10. Compute the actual effective depth and analyze the beam.

## *4.3 Column*

### 4.3.1 Introduction

Columns support primarily axial load but usually also some bending moments. The combination of axial load and bending moment defines the characteristic of column and calculation method. A column subjected to large axial force and minor moment is design mainly for axial load and the moment has little effect. A column subjected to significant bending moment is designed for the combined effect. The ACI Code assumes a minimal bending moment in its design procedure, although the column is subjected to compression force only. Compression force may cause lateral bursting because of the low-tension stress resistance. To resist shear, ties or spirals are used as column reinforcement to confine vertical bars. The complexity and many variables make hand calculations tedious which makes the computer-aided design very useful.

### 4.3.2 Types of Columns

Reinforced concrete columns are categorized into five main types; rectangular tied column, rectangular spiral column, round tied column, round spiral column, and columns of other geometry (Hexagonal, L-shaped, T-Shaped, etc).

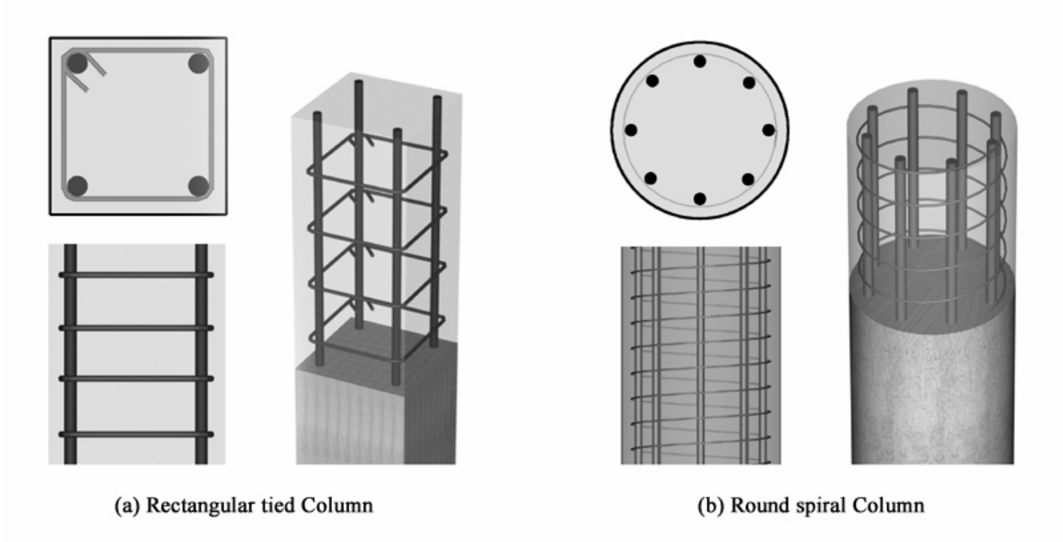



Fig. 4-5: Column types

Fig. 4-5 shows the rectangular tied and round spiral concrete column. Tied columns have horizontal ties to enclose and hold in place longitudinal bars. Ties are commonly No. 3 or No.4 steel bars. Tie spacing should be calculated with ACI Code.

Spiral columns have reinforced longitudinal bars that are enclosed by continuous steel spiral. The spiral is made up of either large diameter steel wire or steel rod and formed in the shape of helix. The spiral columns are slightly stronger than tied columns.

The columns are also categorized into three types by the applied load types; The column with small eccentricity, the column with large eccentricity (also called eccentric column) and biaxial bending column. Fig 4-6 shows the different column types depending on applied load.

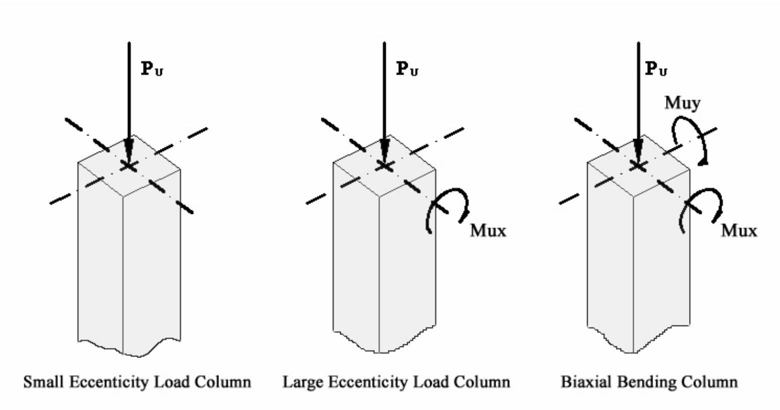



Fig. 4-6: The column types depending on applied load.

Eccentricity is usually defined by location:

- Interior columns usually have
- Exterior columns usually have large eccentricity
- Corner column usually has biaxial eccentricity.

But eccentricity is not always decided by location of columns. Even interior columns can be subjected by biaxial bending moment under some load conditions Fig. 4-7 shows some examples of eccentric load conditions.

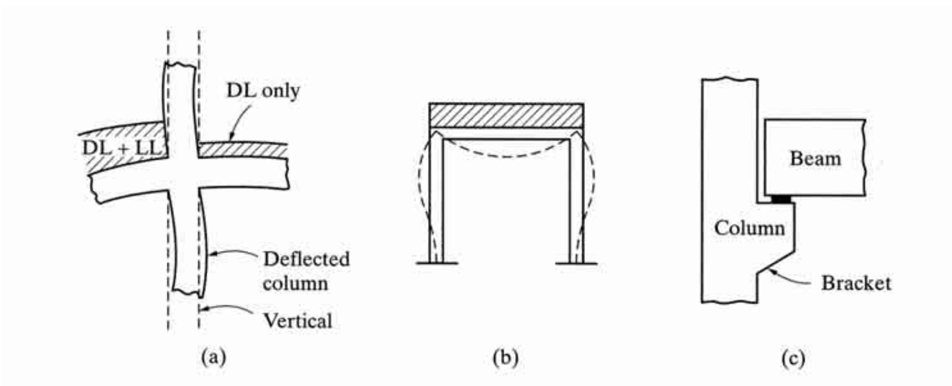



Fig. 4-7: Eccentric loaded conditions (Spiegel, 1998)

### 4.3.3 Design Procedures

- Short Columns with small eccentricities
  1. Establish the material strength and steel area.
  2. Compute the factored axial load.
  3. Compute the required gross column area.
  4. Establish the column dimensions.
  5. Compute the load on the concrete area.
  6. Compute the load to be carried by the steel.
  7. Compute the required steel area.
  8. Design the lateral reinforcing (ties or spiral).
  9. Sketch the design.

- Short Columns with large eccentricities
  1. Establish the material strength and steel area.
  2. Compute the factored axial load ($P_u$) and moment ($M_u$).
  3. Determine the eccentricity ($e$).
  4. Estimate the required column size based on the axial load and 10% eccentricity.
  5. Compute the required gross column area.
  6. Establish the column dimensions.
  7. Compute the ratio of eccentricity to column dimension perpendicular to the bending axis.

8. Compute the ratio of a factored axial load to gross column area.
9. Compute the ratio of distance between centroid of outer rows of bars to thickness of the cross section, in the direction of bending.
10. Find the required steel area using the ACI chart.
11. Design the lateral reinforcing (ties or spiral).
12. Sketch the design.

## *4.4 Footing*

### 4.4.1 Introduction

The foundation of a building is the part of a structure that transmits the load to ground to support the superstructure and it is usually the last element of a building to pass the load into soil, rock or piles. The primary purpose of the footing is to spread the loads into supporting materials so the footing has to be designed not to be exceeded the load capacity of the soil or foundation bed. The footing compresses the soil and causes settlement. The amount of settlement depends on many factors. Excessive and differential settlement can damage structural and nonstructural elements. Therefore, it is important to avoid or reduce differential settlement. To reduce differential settlement, it is necessary to transmit load of the structure uniformly. Usually footings support vertical loads that should be applied concentrically for avoid unequal settlement. Also the depth of footings is an important factor to decide the capacity of footings. Footings must be deep enough to reach the required soil capacity.

### 4.4.2 Types of Footings

The most common types of footing are strip footings under walls and single footings under columns.

Common footings can be categorized as follow:

1. Individual column footing (Fig4-8a)

    This footing is also called isolated or single footing. It can be square, rectangular or circular of uniform thickness, stepped, or sloped top. This is one of the most economical types of footing. The most common type of individual column footing is square of rectangular with uniform thickness.

2. Wall footing (Fig4-8b)

    Wall footings support structural or nonstructural walls. This footing has limited width and a continuous length under the wall.

3. Combined footing (Fig4-8e)

    They usually support two or three columns not in a row and may be either rectangular or trapezoidal in shape depending on column. If a strap joins two isolated footings, the footing is called a cantilever footing.

4. Mat foundation (Fig4-8f)

    Mats are large continuous footings, usually placed under the entire building area to support all columns and walls. Mats are used when the soil-bearing capacity is low, column loads are heavy, single footings cannot be used, piles are not used, or differential settlement must be reduced through the entire footing system.

5. Pile footing (Fig4-8g)

    Pile footings are thick pads used to tie a group of piles together and to support and transmit column loads to the piles.

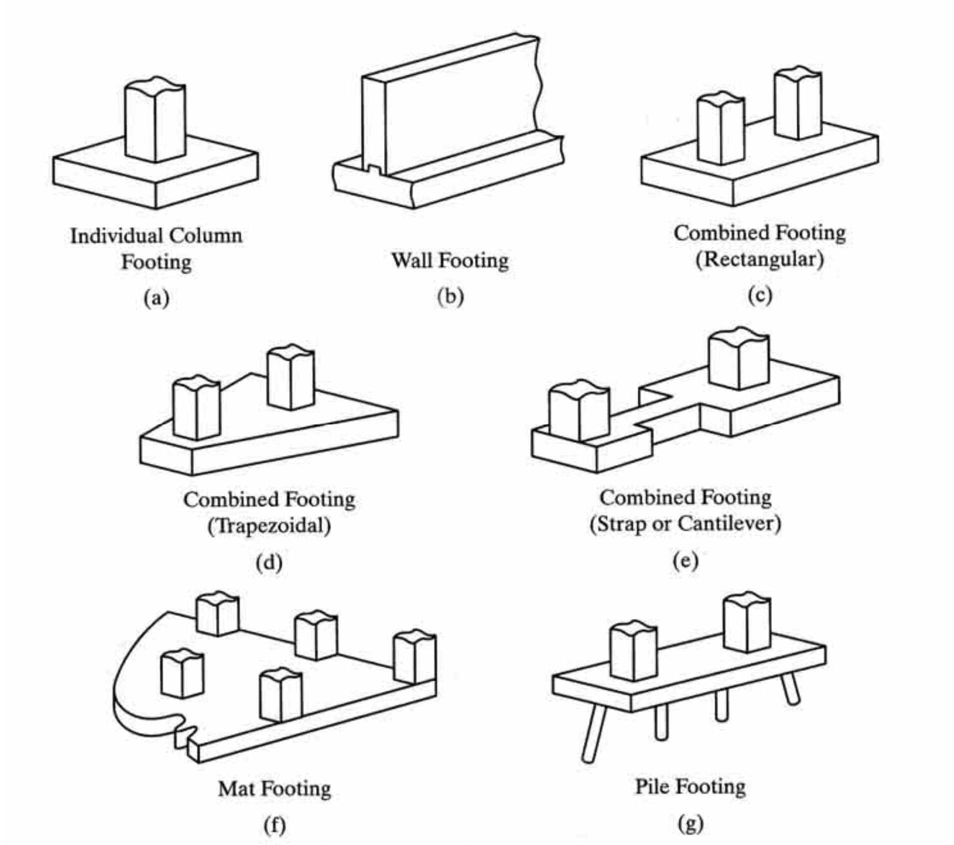


Fig 4-8: Footing types (Spiegel, 1998)

### 4.4.3 Design Procedure

- Wall footing

1. Compute the factored loads.
2. Assume the total footing thickness.
3. Compute the footing self-weight, the weight of earth on top of the footing.
4. Compute the effective allowable soil pressure for superimposed service loads.

5. Determine the soil pressure for strength design.
6. Compute the required footing width.
7. Assume the effective depth for the footing and shear check.
8. Compute the maximum factored moment.
9. Compute the required area of tension steel.
10. Check the ACI Code minimum reinforcement requirement.
11. Check the development length.

- Individual column footing

1. Compute the factored loads.
2. Assume the total footing thickness.
3. Compute the footing self-weight, the weight of earth on top of the footing.
4. Compute the effective allowable soil pressure for superimposed service loads.
5. Compute required footing area.
6. Compute the factored soil pressure from superimposed loads.
7. Assume the effective depth for the footing.
8. Check the punching shear and beam shear.
9. Compute the design moment at the critical section.
10. Compute the required steel area.
11. Check the ACI Code minimum reinforcement requirement.
12. Check the development length.
13. Check the concrete bearing strength at the base of the column.

## Part II: Reinforced Concrete Structure Designer (RCSD)

## 5. Introduction

RCSD is a computer program for reinforced concrete structure design according to the ACI Code. It includes slab, beam, column, and footing design. Its main purpose is to help architecture students who do not have enough structural background but need a structural calculation to design their building. So this program is developed with easy to use interface based on ACI Code procedures.

RCSD provides step by step calculations and is composed of separate modules for beam, slab, column and footing design. The step by step design method is considered one of the best methods to help beginning users, like architecture students. For example, users do not need to input the all required data at once. The program asks the minimum required data and provides default-input data. The user can use the default data or select other data.

The modular RCSD program structure also has the advantage that each module is executable separately and the user can add other modules.

RCSD is programmed using Microsoft Visual Basic version 6.0. Visual Basic is much easier to learn than other languages and provides good graphic user interface (GUI). Each module is composed of multiple pages that have been organized using Microsoft Tabbed Control Dialog Component. Each module is executed step by step along the tabs. Tabs are divided into frames for better organization of different category of input and output data.

# 6. Slab Module

## *6.1 Introduction*

RCSD supports two different types of slab: One-way solid slabs and two-way slabs. One-way slabs are assumed as rectangular beams of 12inch width. One-way slabs are assumed to span the short direction analized as beam-like strips of unit width. Design of two-way slabs is more complex than one-way slabs. The two-way slab design module assumes the minimum slab thickness according to the ACI Code and calculates the deflection based on applied service loads. The two-way slab design module defines the approximate slab thickness rather than detail calculations.

## *6.2 One-way solid slab design module*

RCSD designs the one-way slab in the sequence of INPUT, MOMENT, REINFORCEMENT, CHECK, and DRAWING tabs.

The INPUT tab requests: Load Condition, Material Strength, and Dimensions. The Load Condition frame requests two items: dead load and live load. RCSD includes small assistant programs to help user input and is executed by clicking the “ASSIST” button located on the frames.

The Load Condition frame also has an assistant program for dead load input called Dead Load Calculator (Fig. 6-1). This assistant program calculates the dead load by just checking the material. It will return the total dead load to main program.

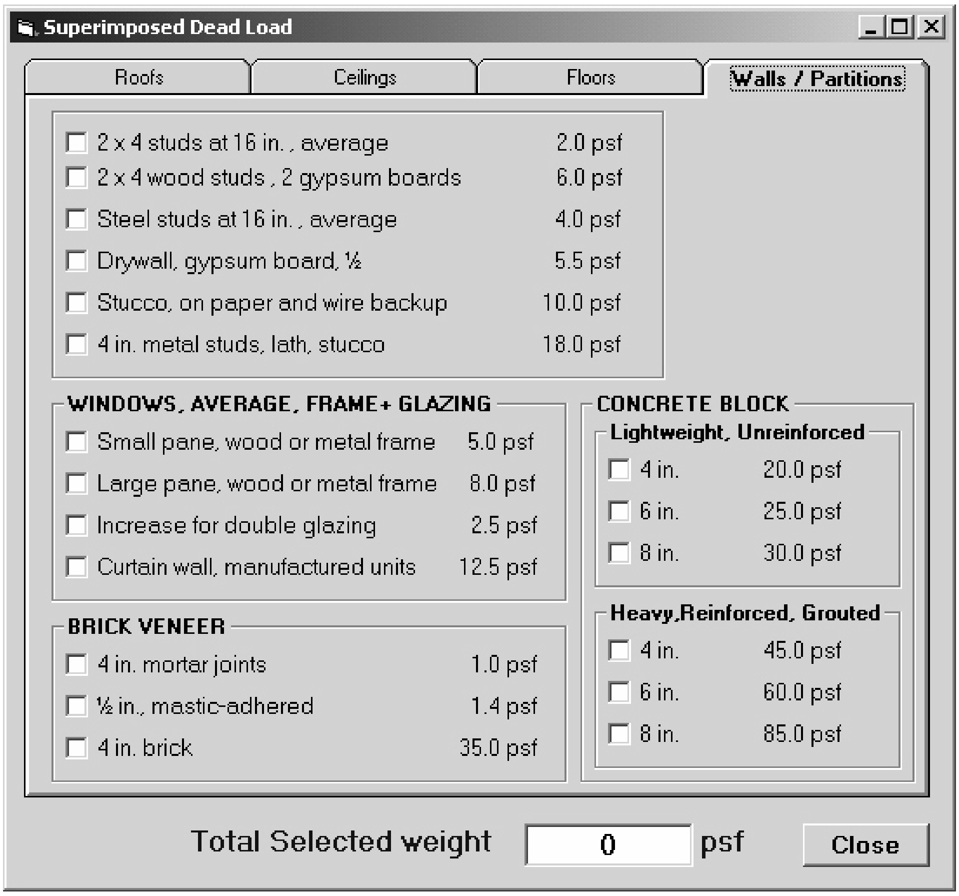


Fig. 6-1: Superimposed Dead Load Calculator

Live load input assistant button activates the building usage list box, which automatically assigns required uniformly distributed live load when the user selects the building usage from the list box.

In MOMENT tab, RCSD calculates the minimum slab thickness, factored load, and moments using the previous input data. The minimum thickness of one-way slabs is calculated using the minimum thickness of non-prestressed one-way slabs from ACI Code. Slab design module will design the slab based on this thickness including immediate and long-term deflection calculations. After calculating the minimum thickness, RCSD computes the slab self weight and add into superimposed dead load and calculate factored total load. The five different moments of slab can be obtained from MOMENT tab with graph and slab shape (Fig. 6-2).

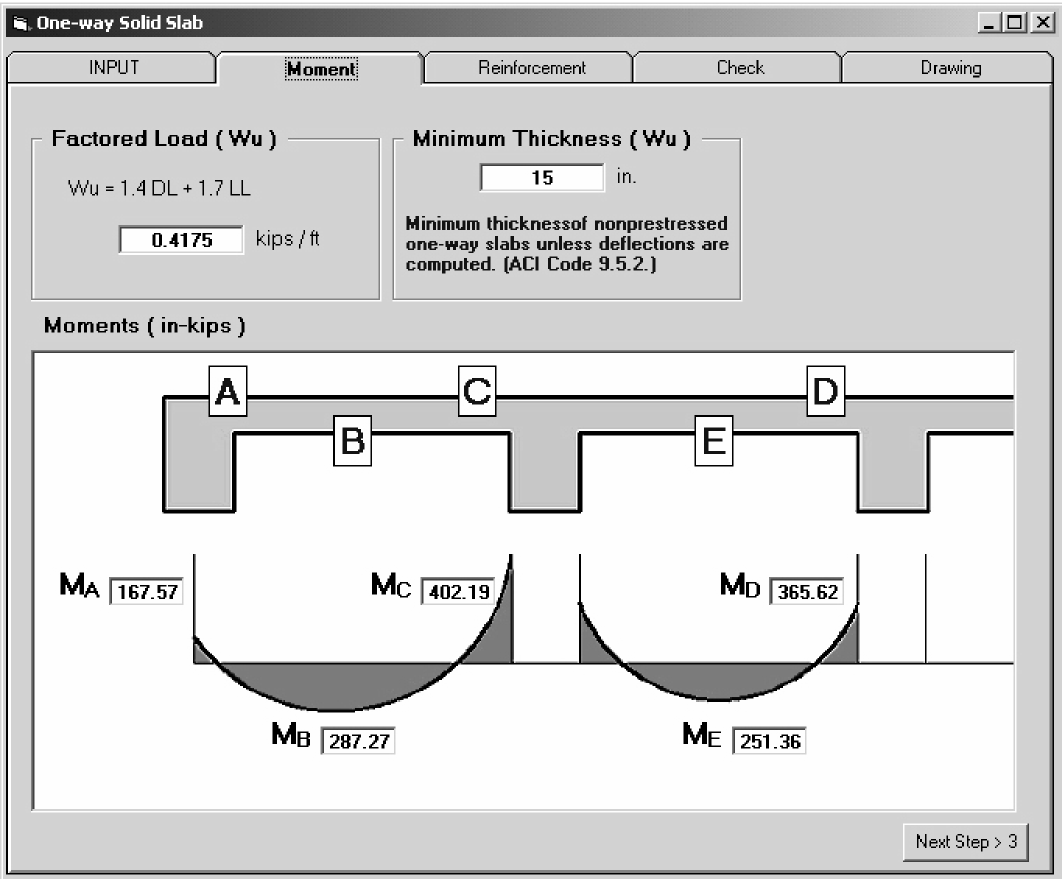


Fig. 6-2: MOMENT tab of One-way slab module

The required steel area is calculated according to the moments. RCSD asks the user to select the bar size from the steel bar list box in each frame. The steel bar list box shows steel bars and cross section area of each bar. When the user selects the steel bars, RCSD draws the steel bars into slab section and calculates the necessary spacing (Fig. 6-3). Steel bar spacing is based on maximum possible spacing defined by the ACI Code, section 7.6.5.

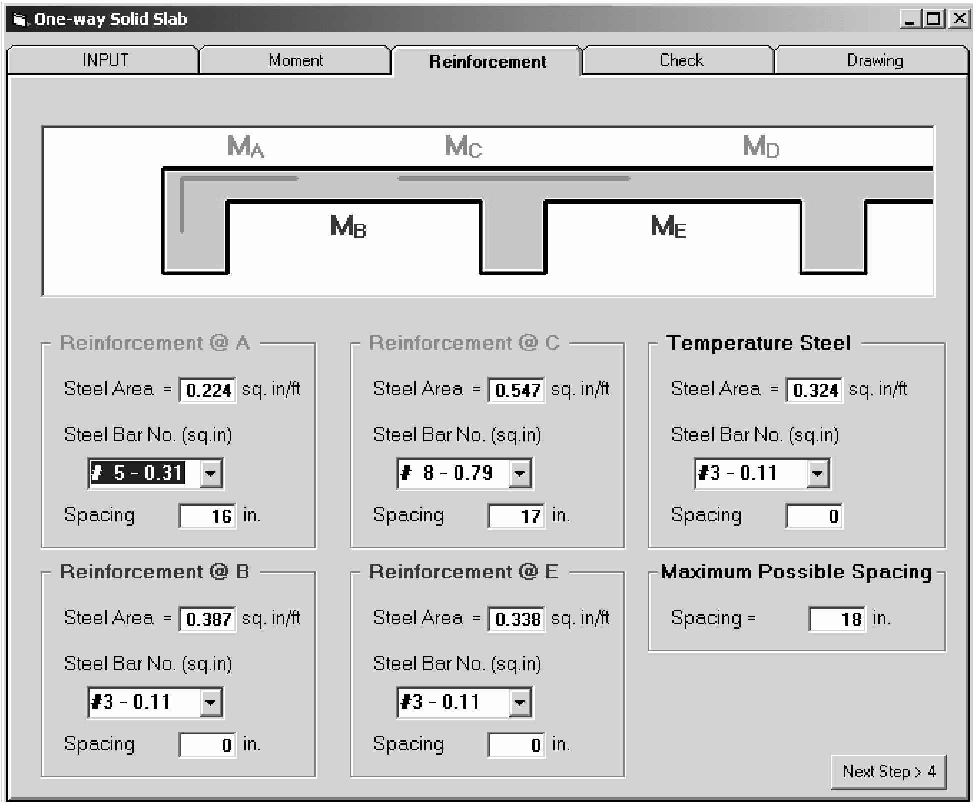



Fig. 6-3: REINFORCEMENT tab of One-way slab module

After the bending design process, RCSD compares shear strength of critical sections with the shear strength of the concrete. Also it checks whether *Current design thickness* is greater than the *Min. Thickness by ACI Code* and gives a warning massage if it is less and provide return routine to change the slab thickness.

Elastic and long-term creep deflections are computed, using the effective moment of inertia and steel area. RCSD also checks the two deflections versus maximum permissible deflection (Fig. 6-4).

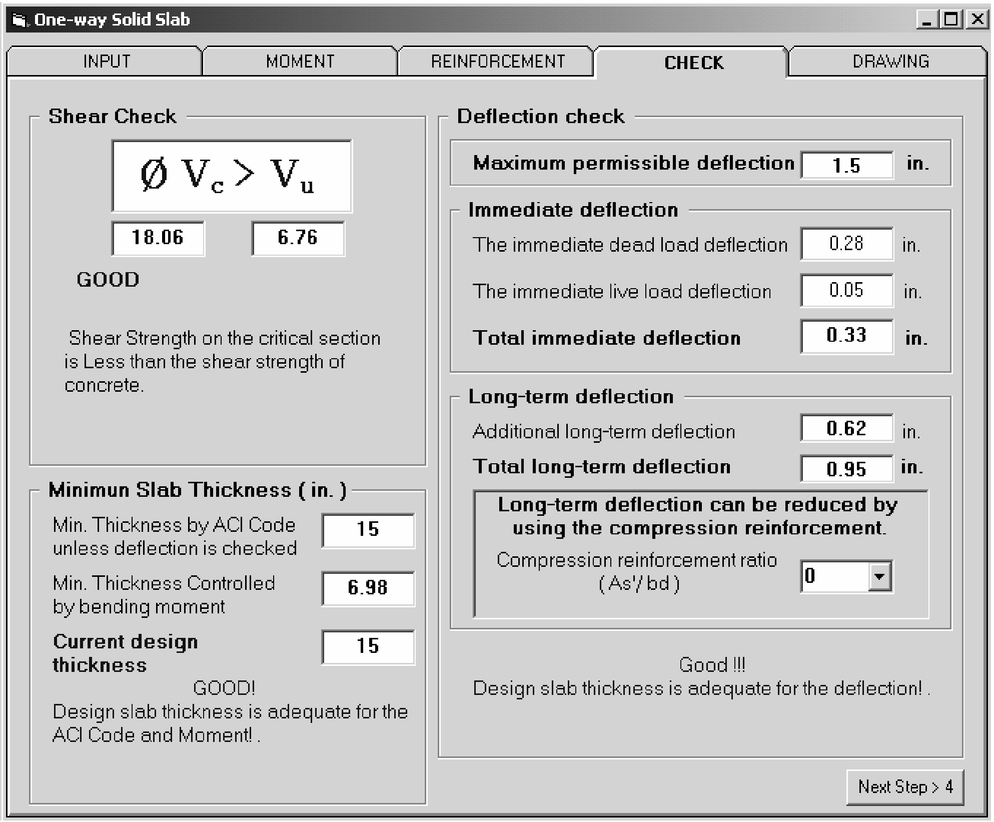


Fig. 6-4: Shear and deflection checks of One-way slab module

### *6.3 Two-way slab design module*

The two-way slab design module is composed of four tabs: INPUT, SHAPE and SIZE, THICKNESS, and DEFLECTION.

The two-way slab design module has similar procedures as the one-way slab but it designs the minimum slab thickness using preliminary design thickness of the ACI Code. The program requests the slab type input and then outputs the minimum thickness and draws the chart to compare different conditions. Fig. 6-5 shows the THICKNESS tab of the two-way slab module.

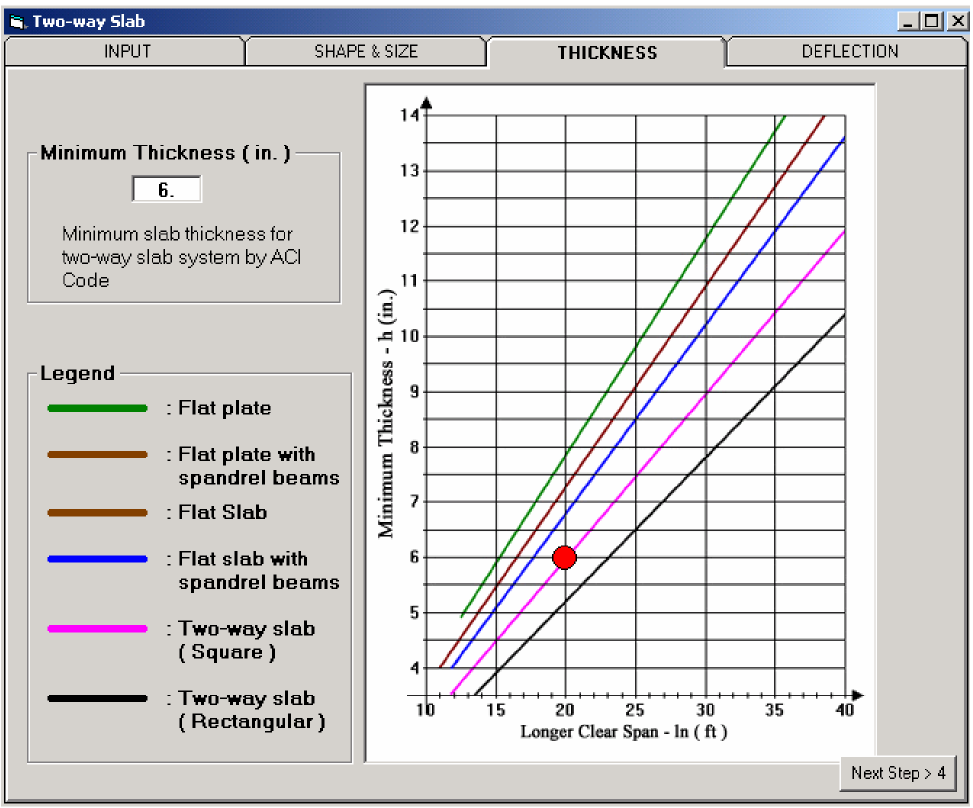



Fig. 6-5: Two-way slab minimum thickness output

To check the deflection of two-way slabs, RCSD uses the simplified analysis method. Deflection is checked based on service load. The applied loads are reduced to reduction factor and the half of the maximum possible steel area is used to compute deflection. Also, the user can check change of deflection by inputting new compression and tension steel ratio (Fig. 6-6).

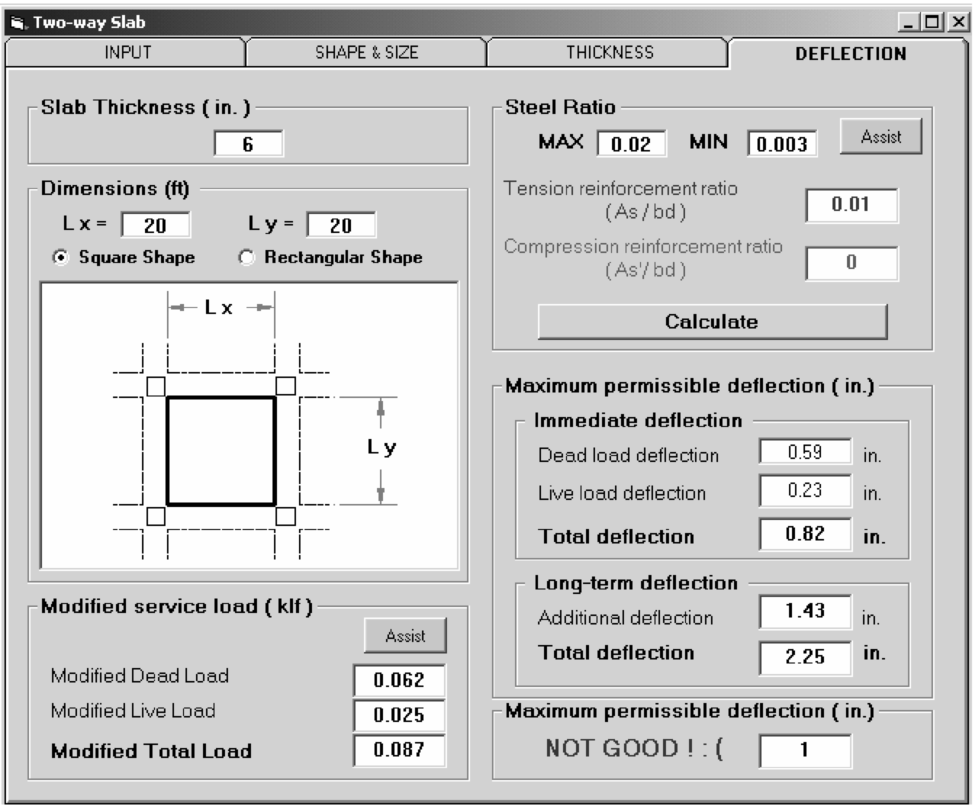
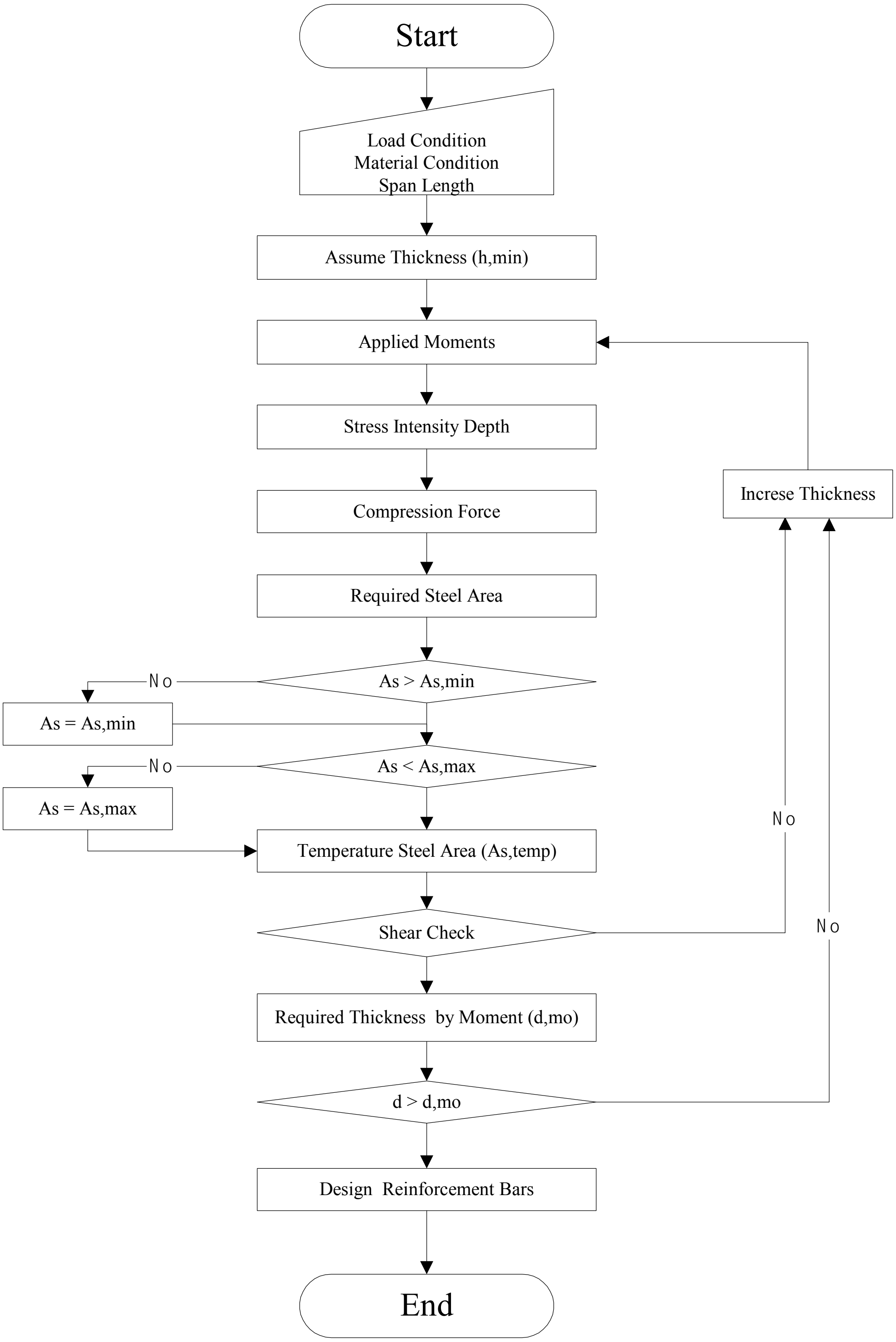


Fig. 6-6: Two-way slab deflection check tab

## *6.4 Flow Chart*



# 7. Beam Module

## *7.1 Introduction*

RCSD provides single and double reinforced beam design method in one module. The beam design module has nine tabs: INPUT, SIZE, TYPE, STEEL AREA, BAR DESIGN, SHEAR, DEFLECTION, and DATA SHEET tab.

## *7.2 Beam design module*

RCSD calculates the minimum thickness of the beam using the minimum thickness of non-prestressed beams according to ACI Code 9.5.2. The INPUT tab requests to input the tributary area data and connection type (Fig. 7-1).

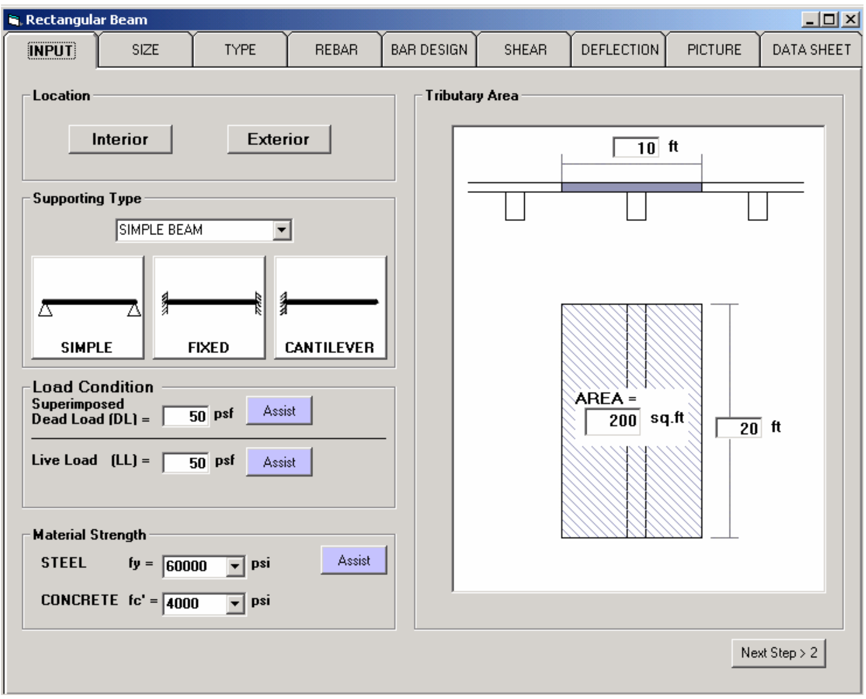



Fig. 7-1: Beam tributary area and support type input

RCSD asks for applied load and material used in slab design module. According to the input data, it calculates the beam size, reinforcement type and required steel area. In the BAR DESIGN tab, users can select the steel bar size, similar to the reinforcement design of the one-way slab design module. RCSD calculates the required number of steel rebars and spacing, and shows a scaled drawing of beam size and reinforcement (Fig. 7-2).

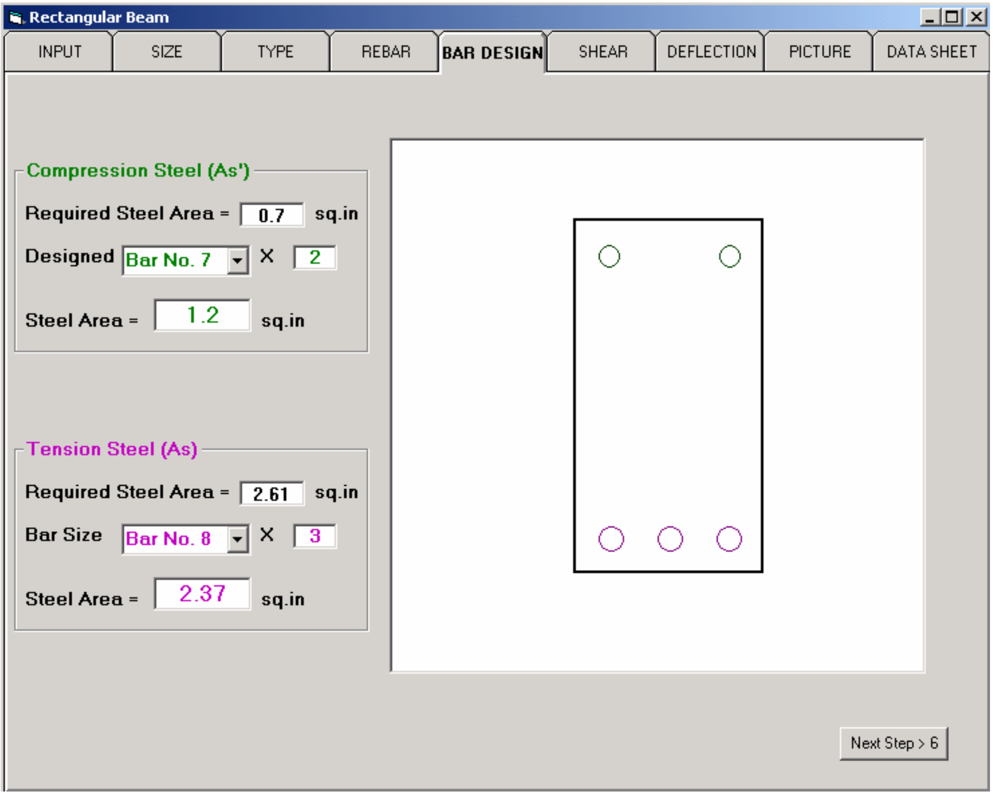



Fig.7-2: Beam reinforcement design module

The SHEAR tab compares the shear force at the critical section and unreinforced concrete shear capacity. If the beam requires shear reinforcing, RCSD provides the stirrup design routine to help the user select stirrup size and maximum spacing (Fig. 7-3).

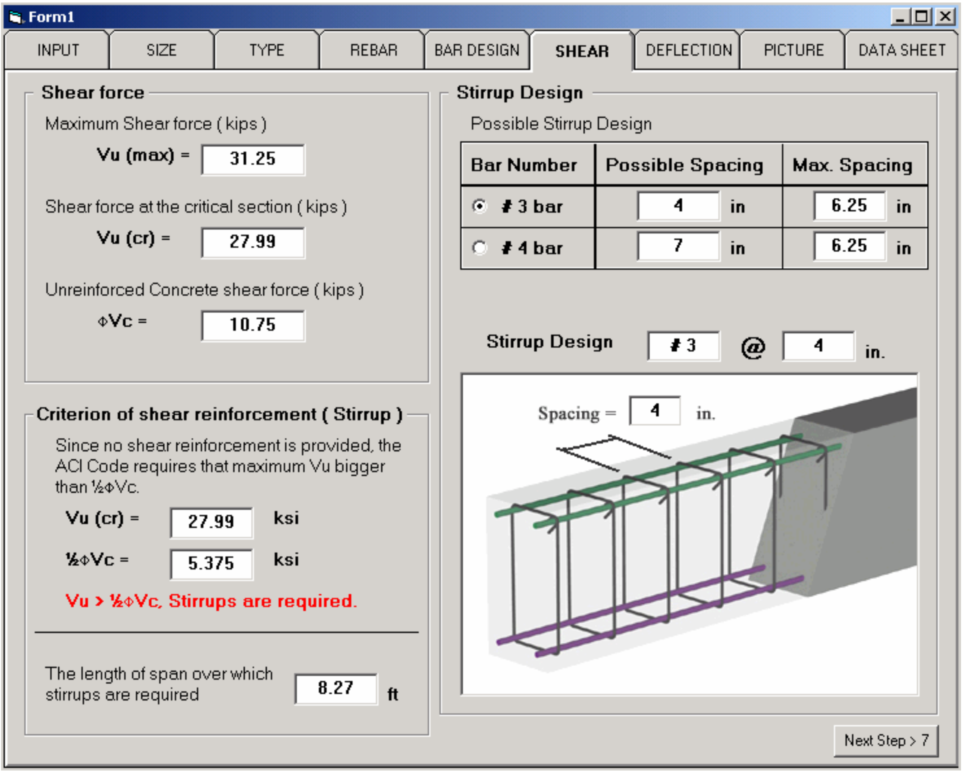
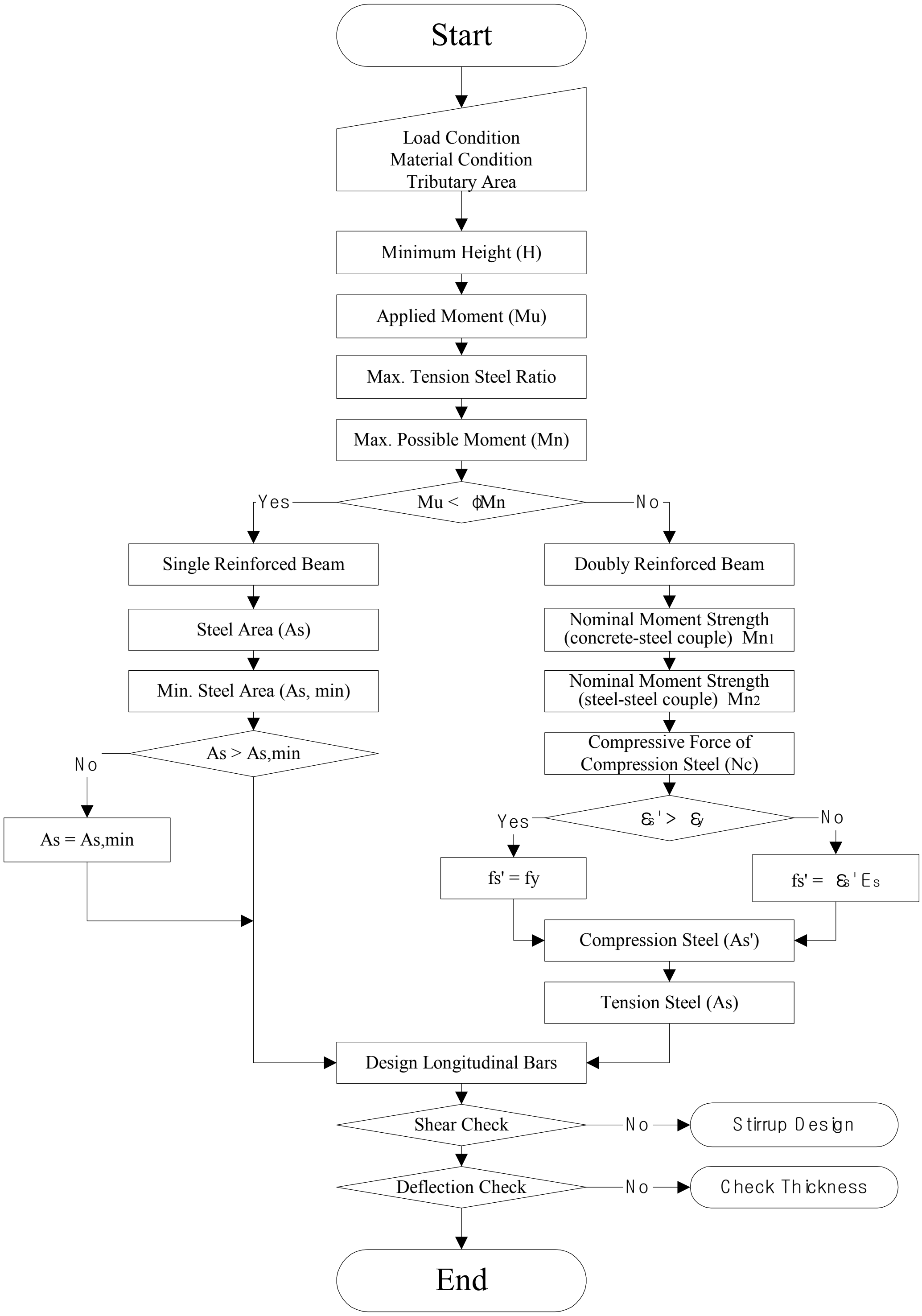
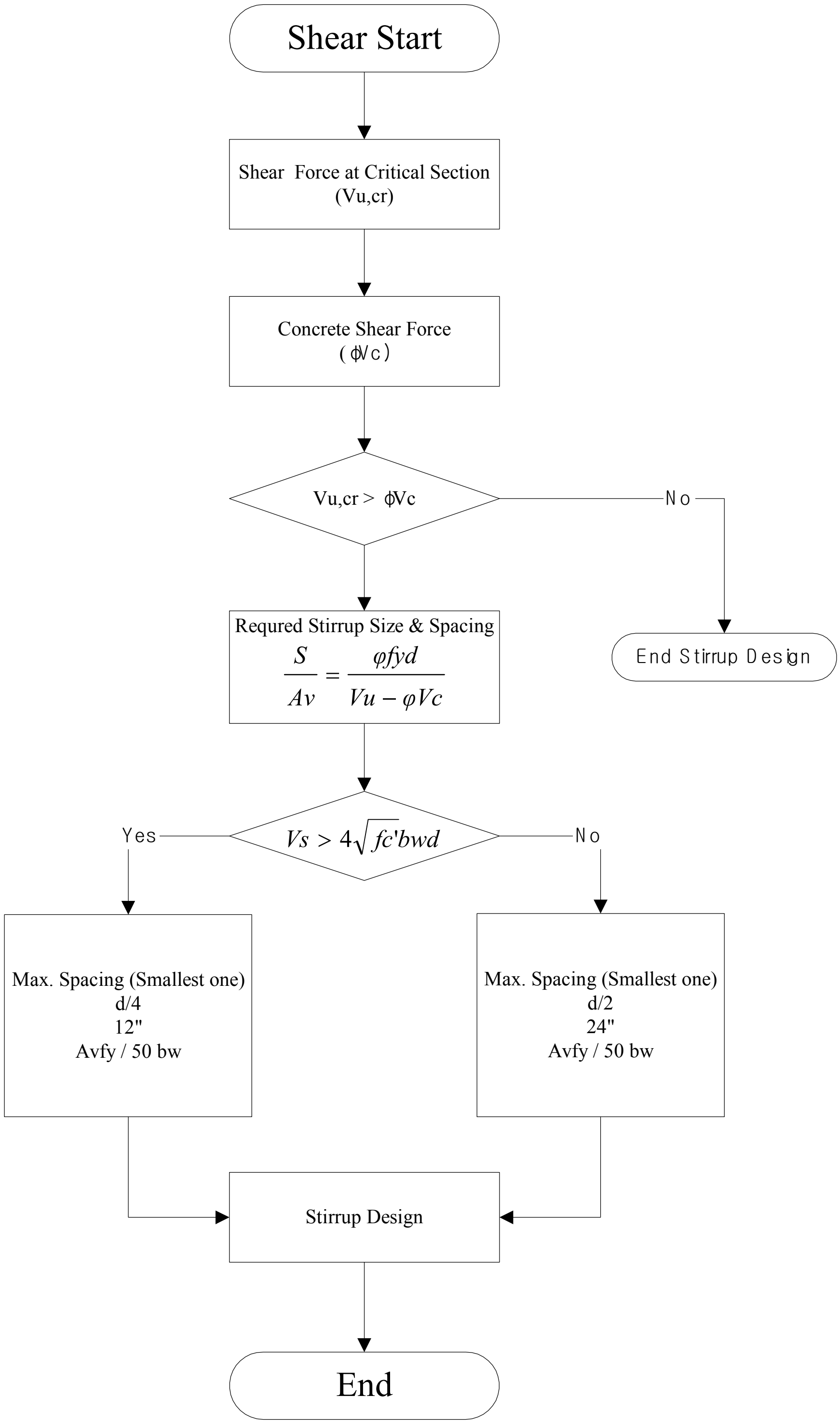



Fig. 7-3: The shear check and stirrup design

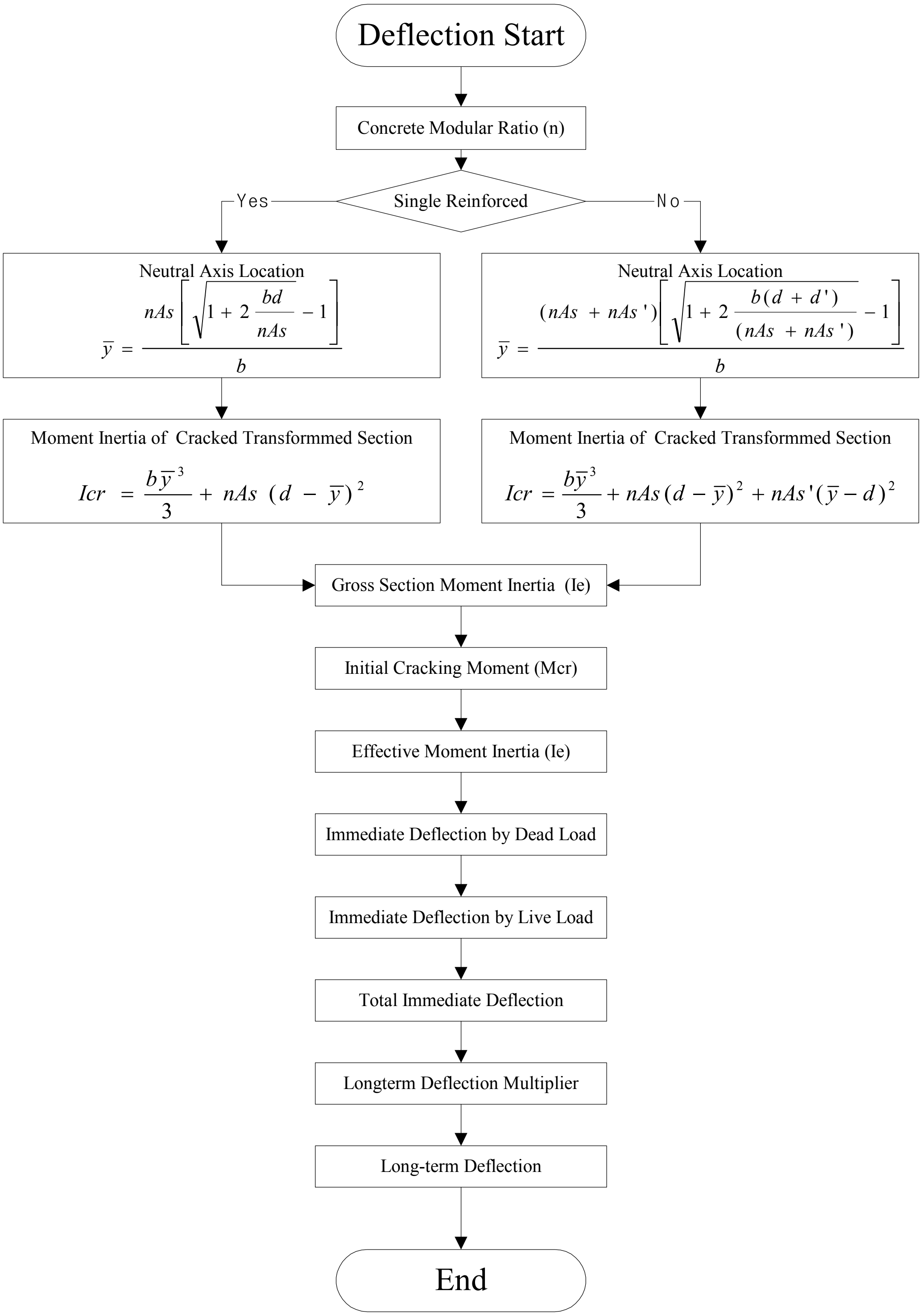
## *7.3 Flow Chart of Beam Design*



## 7.4 Flow Chart of Shear Check



### 7.5 Flow Chart of Deflection Check



# 8. Column Module

## *8.1 Introduction*

Column design can be categorized into three different types according to applied load: column with small eccentricity, column with large eccentricity and column with biaxial bending. RCSD provides first two types of column design. The design of column carrying small eccentricity is calculated by simple method, computed by the ACI method for axial load with small eccentricity. If the axial load is applied with eccentricity, the column is subjected to moment and needs more bending strengths. When the bending moment increases, its axial load strength decreases. The relation between axial strength and bending strength varies according to eccentricity, steel ratio, concrete cover, and material strength.

The P-M interaction diagram shows the relationship of axial load strength and bending moment (Fig. 8-1).

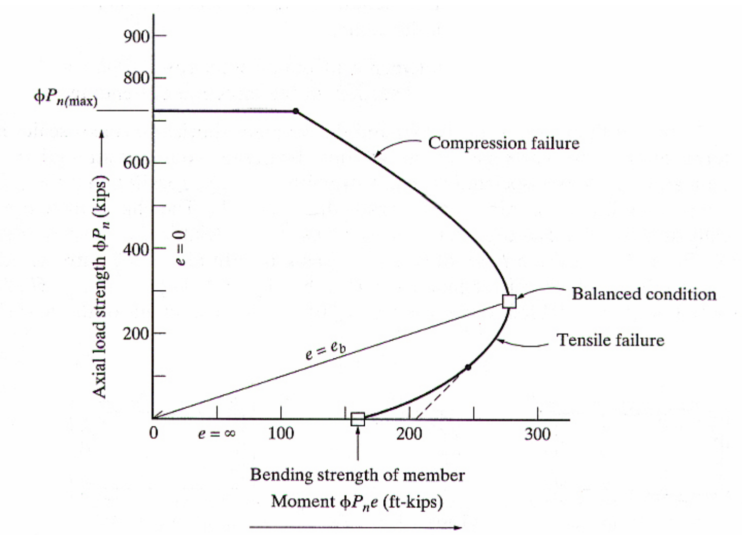



Fig. 8-1: Column interaction diagram (Spiegel, 1998)

The P-M interaction diagram has three different condition zones: balanced condition, compression failure, and tensile failure condition. The American Concrete Institute (ACI) provides of P-M interaction diagram for various conditions to help design the column. However it would be difficult to use all P-M interaction diagrams in RCSD. A simplified method is needed for the column design module.

### *8.2 Column analysis program*

The ACI Code does not allow tensile failure of columns. This means only the compression failure zone of the P-M interaction diagram is used to design columns. Based on this a small column analysis program was developed (Fig. 8-2).

The Simplified Column Analysis Program (SCAP) has some limitation. The column should be rectangular in shape and reinforcement steel bars are arranged along the small side of column section. SCAP computes the axial load, moment strengths and stresses.

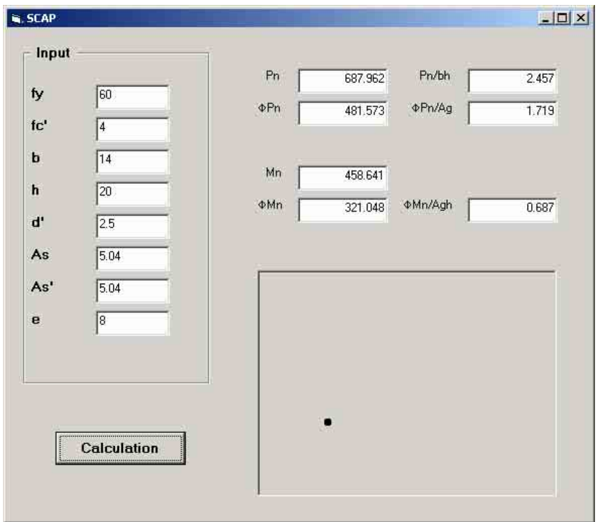



Fig. 8-2: Simplified Column Analysis Program

### *8.3 Simplified P-M interaction diagram*

SCAP simplifies the original ACI P-M interaction diagram. Various column conditions except the concrete and steel condition and concrete cover, are randomly generated, using 4 ksi and 6 ksi concrete, 2.5inch rebar cover and steel ratios from 2% to 5% (most common in the design of column). Fig. 8-3 shows the resulting P-M interaction diagram.

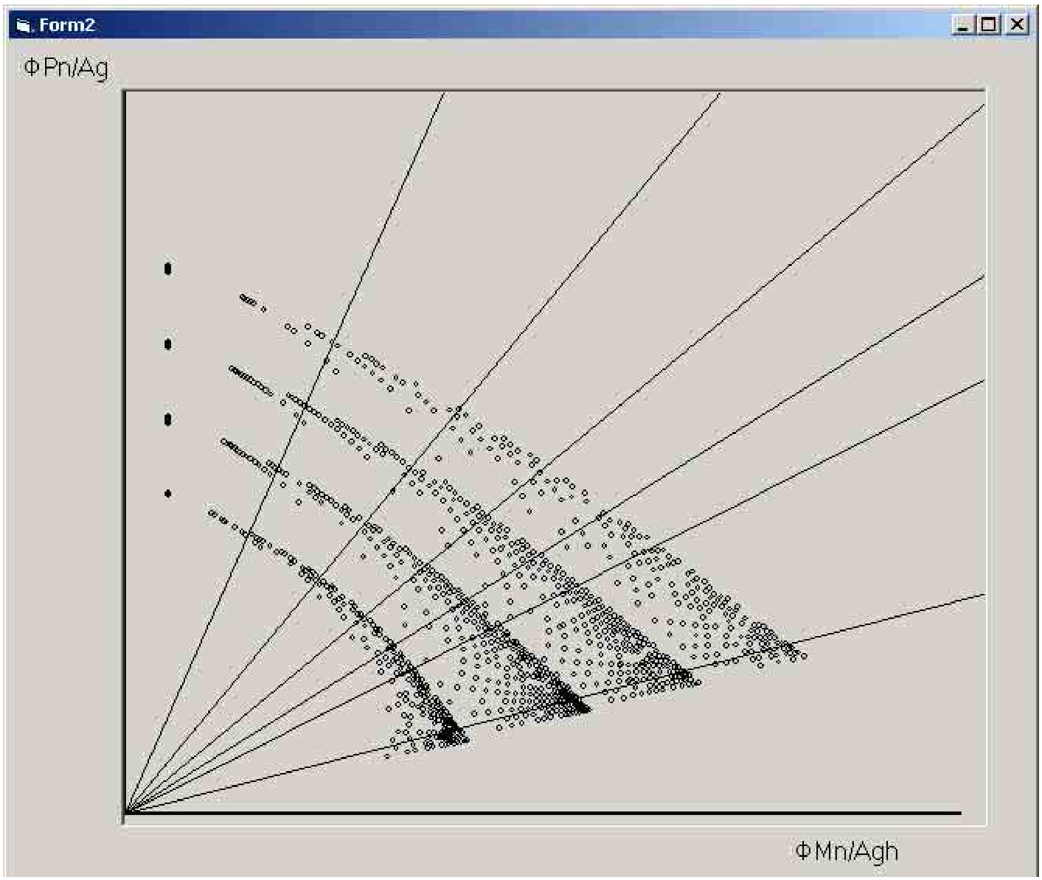



Fig.8-3: SCAP simplified P-M interaction diagram

All P-M data was scattered and it was difficult to find a general equation for the simplified P-M interaction diagram, due to different values for various cover thickness. Random column size and cover thickness resulted in different P-M conditions. To find a general equation, SCAP was modified with fixed concrete cover, which is 15% of the wider column size (Fig. 8-4).

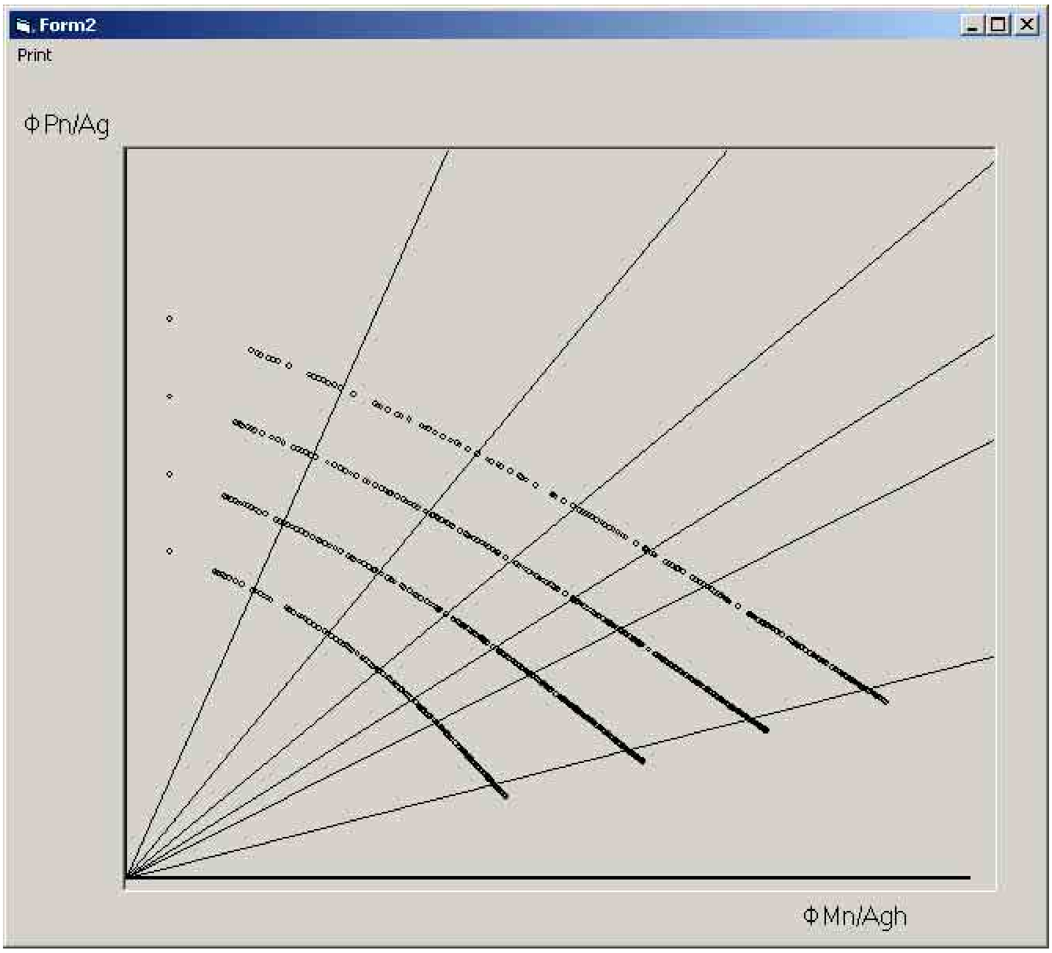



Fig. 8-4: P-M interaction diagram with 15% cover thickness

The P-M interaction diagram with 15% cover showed more consistent data for a general equation for the simplified P-M interaction diagram based on 10,000 random P-M data samples.

### *8.4 General equation for simplified P-M interaction diagram*

The general equation for SPSS is based on statistical analysis for steel ratios from 2% to 5%. Table 8-1 shows the equations for each steel ratio from 2-5%.

| Steel Ratio (%) | Simplified P-M interaction equation |
|---|---|
| 5 % | $\frac{\phi Pn}{Ag} = -0.8 \times \left(\frac{\phi Mn}{Agh}\right)^2 - 1.54 \times \left(\frac{\phi Mn}{Agh}\right) + 4.3$ |
| 4 % | $\frac{\phi Pn}{Ag} = -1.0 \times \left(\frac{\phi Mn}{Agh}\right)^2 - 1.50 \times \left(\frac{\phi Mn}{Agh}\right) + 3.9$ |
| 3 % | $\frac{\phi Pn}{Ag} = -1.33 \times \left(\frac{\phi Mn}{Agh}\right)^2 - 1.46 \times \left(\frac{\phi Mn}{Agh}\right) + 3.5$ |
| 2 % | $\frac{\phi Pn}{Ag} = -2.0 \times \left(\frac{\phi Mn}{Agh}\right)^2 - 1.42 \times \left(\frac{\phi Mn}{Agh}\right) + 3.1$ |

Table 8-1: Simplified P-M interaction equations

Fig 8-5 shows the simplified P-M interaction diagram generated by simplified equations of Table 8-1.

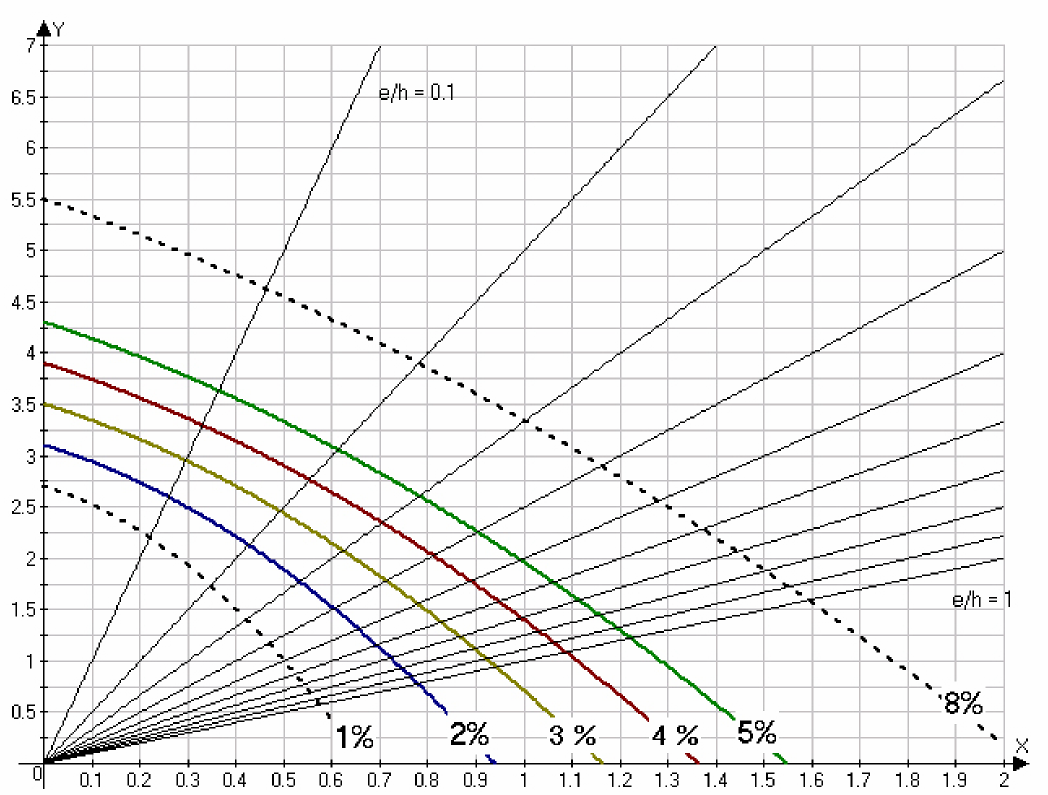



Fig. 8-5: Simplified P-M interaction diagram

The simplified equations are based on concrete of *f'*c =4000 psi and steel of fy = 60000 psi. The relationship between material strength and axial stress checked with SCAP showed that axial stress is directly proportional to material strength.

Fig. 8-6 and Fig. 8-7 correlate axial stress with steel and concrete strength.

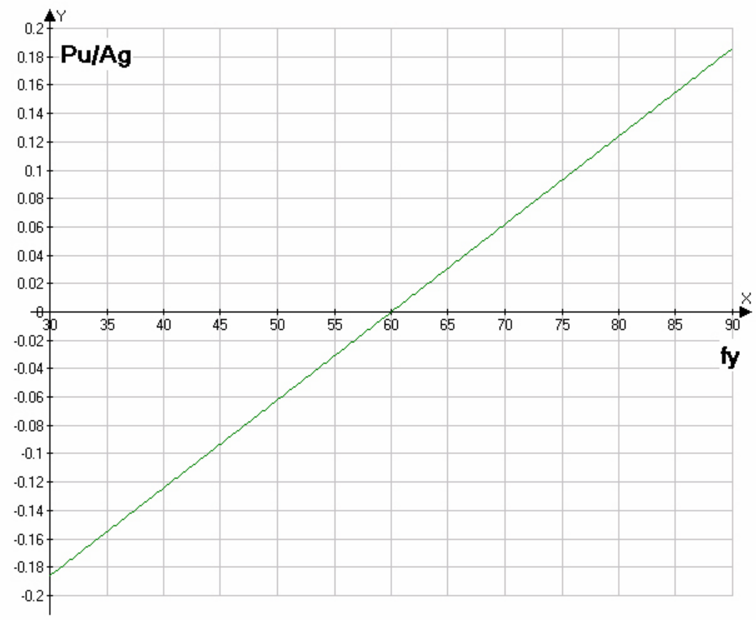



Fig. 8-6: Axial stress and steel strength

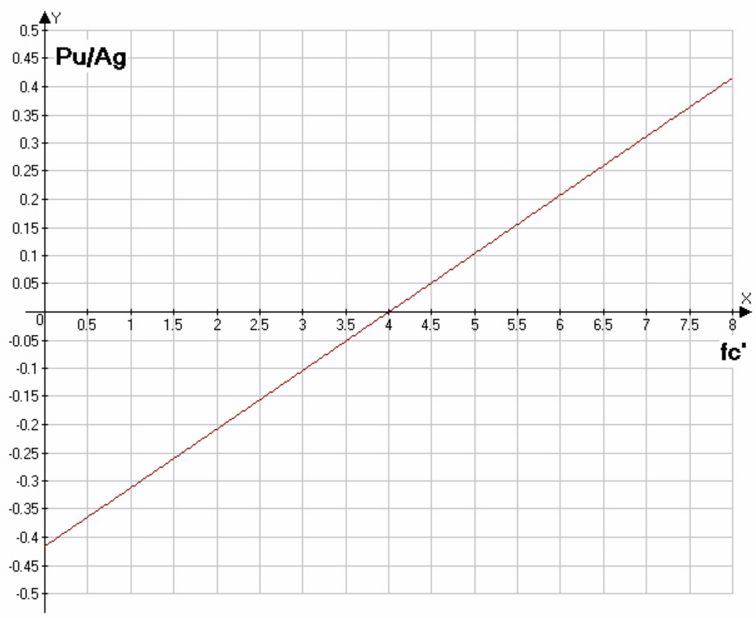



Fig. 8-7: Axial stress and concrete strength.

Considering various concrete and steel strength, the simplified general equation was used to calculate various eccentrically loaded rectangular columns to find approximate steel ratios of the column for column module.

The simplified equation for P-M interaction diagram is

$$\frac{\phi Pn}{Ag} = -\frac{0.04}{R} \times \left(\frac{\phi Mn}{Agh}\right)^2 - (1.34 + 4R) \times \left(\frac{\phi Mn}{Agh}\right) + (2.3 + 40R) + \alpha$$

Where $\alpha = (fy - 60) \times 0.0062 + (fc' - 4) \times 0.104$

SCAP generated another graph for the axial strength reduction factor depending on the ratio of column size and eccentricity, defined as e/h , where e = eccentricity and h = column thickness (Fig.8-8).

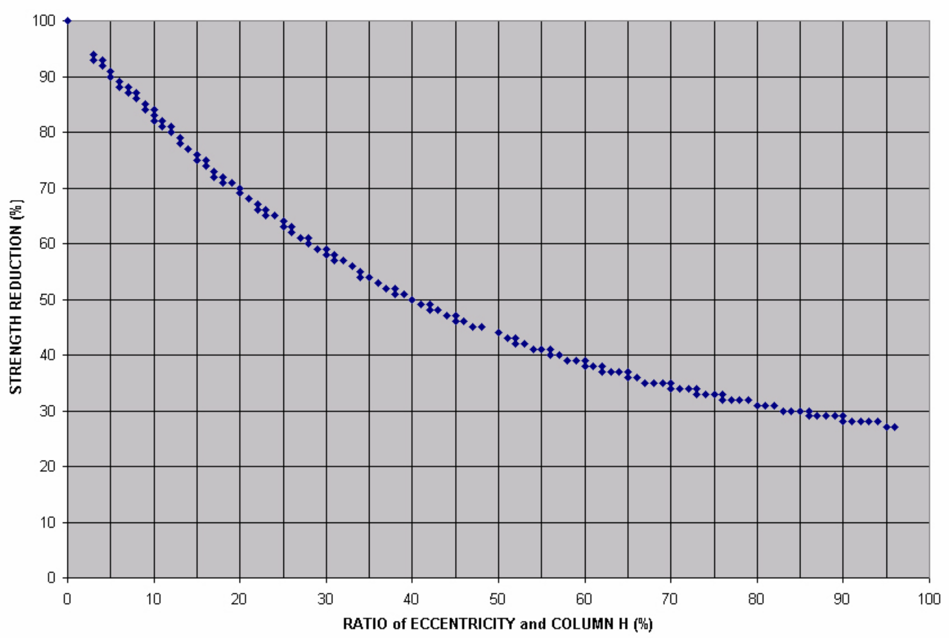


Fig. 8-8: Strength reduction in % due to column moment

### *8.5 Column design module*

The column design module has six tabs: INPUT, LOAD, SIZE, CHECK, REINFORCEMENT, and PICTURE tab.

The INPUT and LOAD tab provide graphical input column type, applied axial load and moments. Based on the axial load and moment, RCSD assumes the column size and calculates the steel area.

When no moment is applied to the column, RCSD calculates the required gross column area with 3% steel area and designs a square column. In case of eccentrically loaded column, it assumes a rectangular column with 1:1.5 section ratio and calculates the ratio of eccentricity to the larger column side and the reduction factor according to the strength reduction graph (Fig 8-7). RCSD increases the required column size considering the reduction factor. To calculate the steel area, the simplified P-M interaction equation is used.

The steel ratio equation is

$$R = \frac{\beta + \sqrt{\beta^2 + 144\gamma}}{72}$$

Where: $\alpha = (fy - 60) \times 0.0062 + (fc' - 4) \times 0.104$

$$\beta = \frac{\phi Pn}{Ag} - 0.96 + \alpha$$

$$\gamma = 0.04 \times \left( \frac{\phi Mn}{Agh} \right)^2$$

In the CHECK tab, RCSD analyzes the designed column and shows the calculation result, compares applied design load and moment, and draws the simplified P-M interaction diagram. (Fig. 8-9)

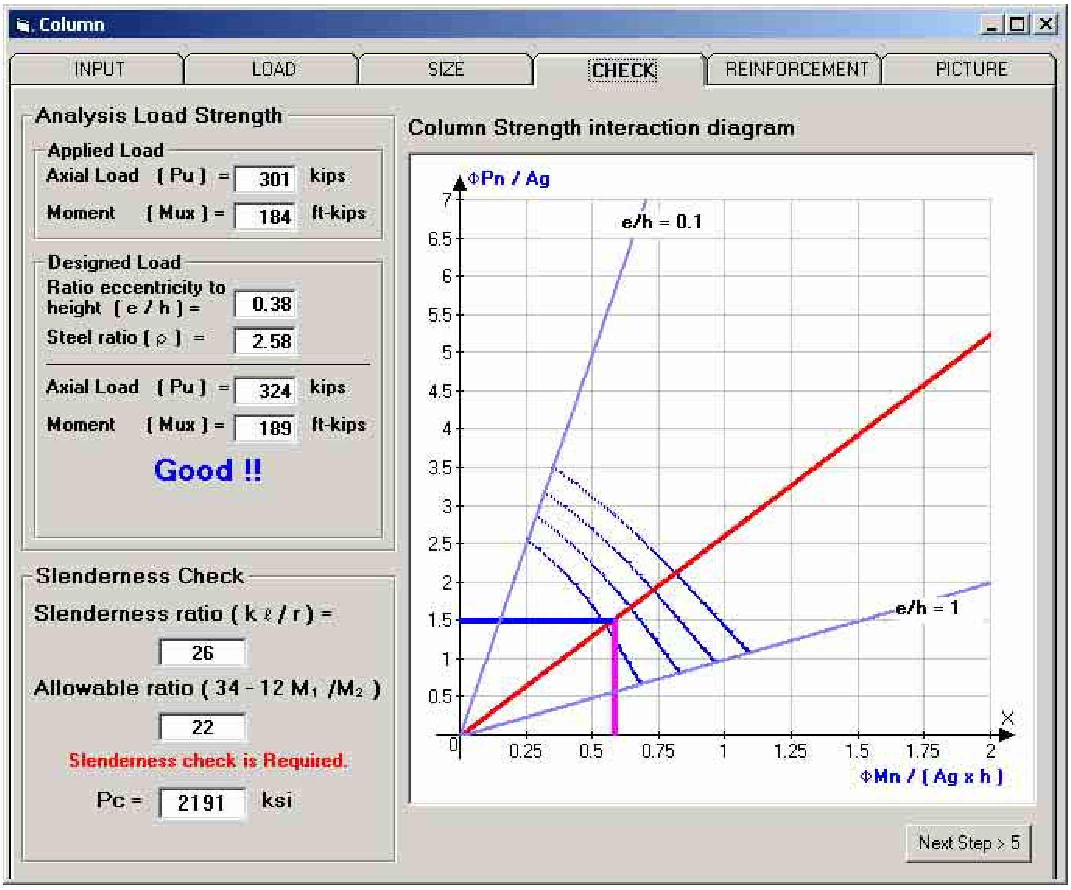
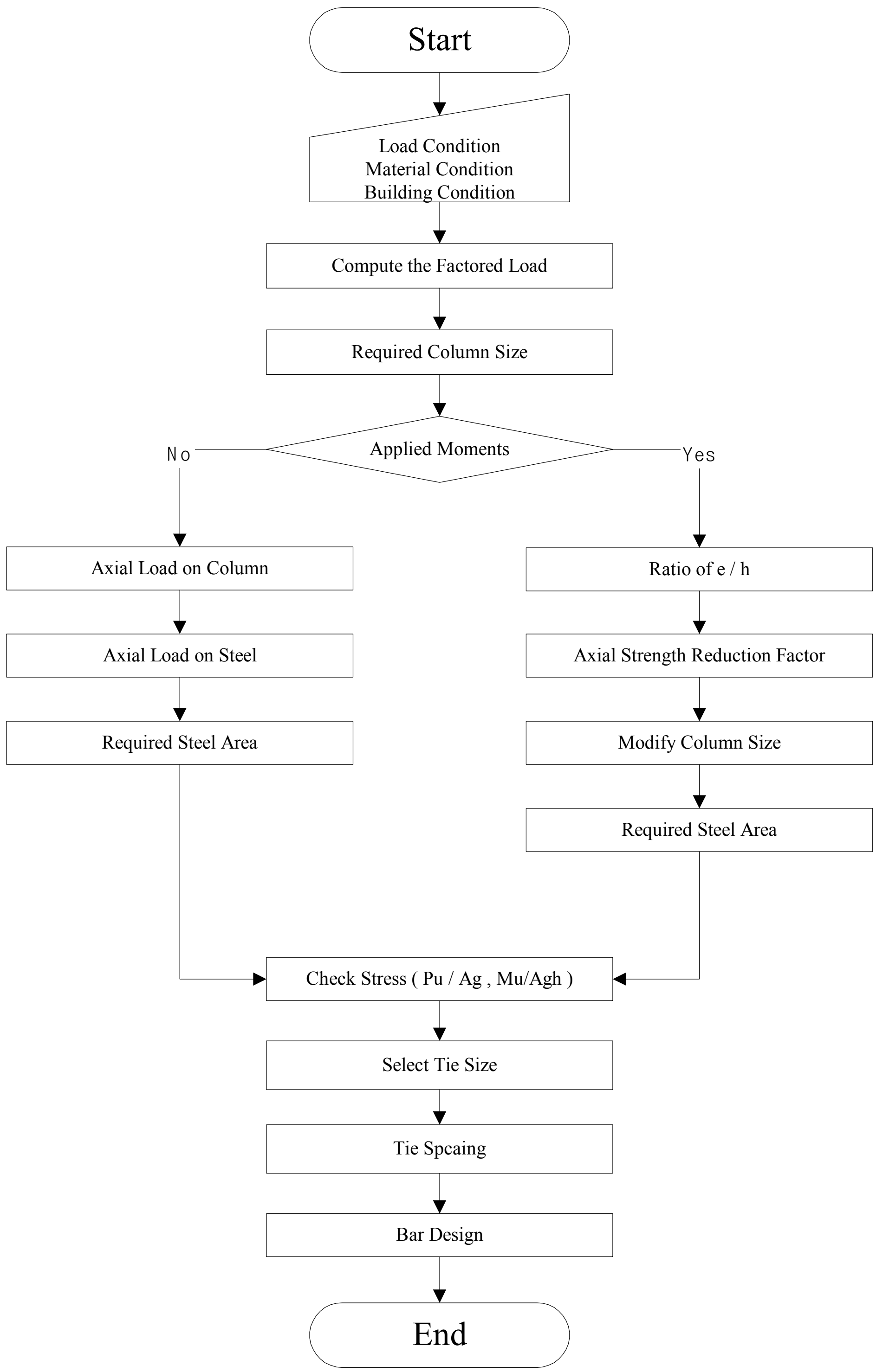



Fig. 8-9: CHECK tab of column design module

From the REINFORCEMENT tab, the user can select the main bar and tie bar sizes. RCSD calculates the required quantities and spacing. It also shows the typical tie arrangements on a 3d image.

### *8.6 Flow Chart*



# 9. Footing Module

## *9.1 Introduction*

RCSD provides two footing design modules for wall footings and individual column footings. Both modules have five tabs: INPUT, SIZE, REINFORCEMENT, DEVELOPMENT and DRAWING. The INPUT tab requests required conditions, such as service load, material strength, and soil conditions.

Based on the input data RCSD calculates possible footing size and thickness to resist shear in the SIZE tab (Fig. 9-1).

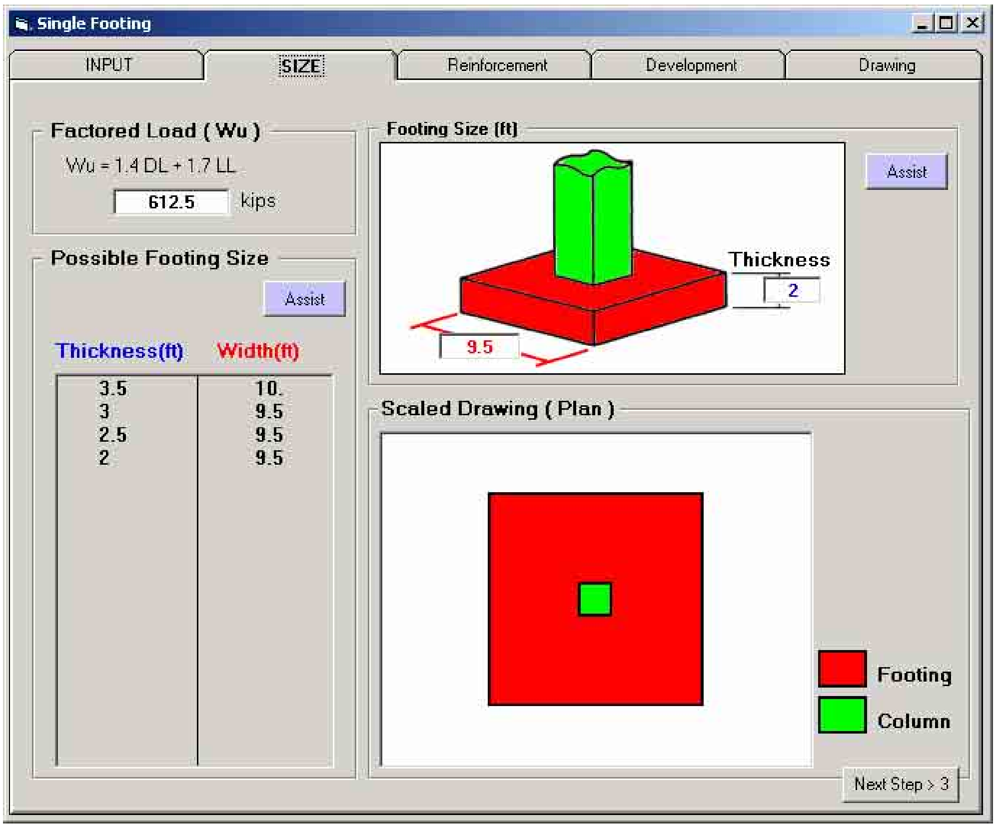



Fig. 9-1: The possible footing size calculation table and drawing

The REINFORCEMENT tab outputs the required steel area to support the moment and allows the user to select the size of steel bar. RCSD computes bar quantities and spacing to draw the scaled footing and its rebars (Fig. 9-2).

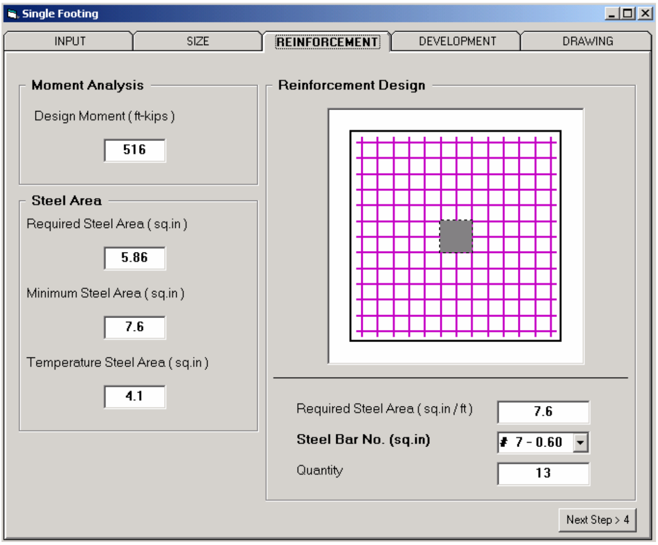



Fig. 9-2: REINFORCMENT tab of individual column footing

It also shows the footing design with plan, elevation and reinforcing in the DRAWING tab (Fig. 9-3).

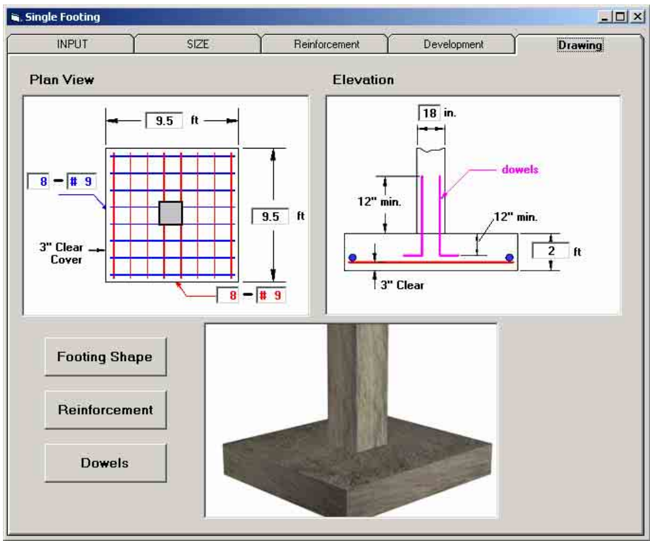
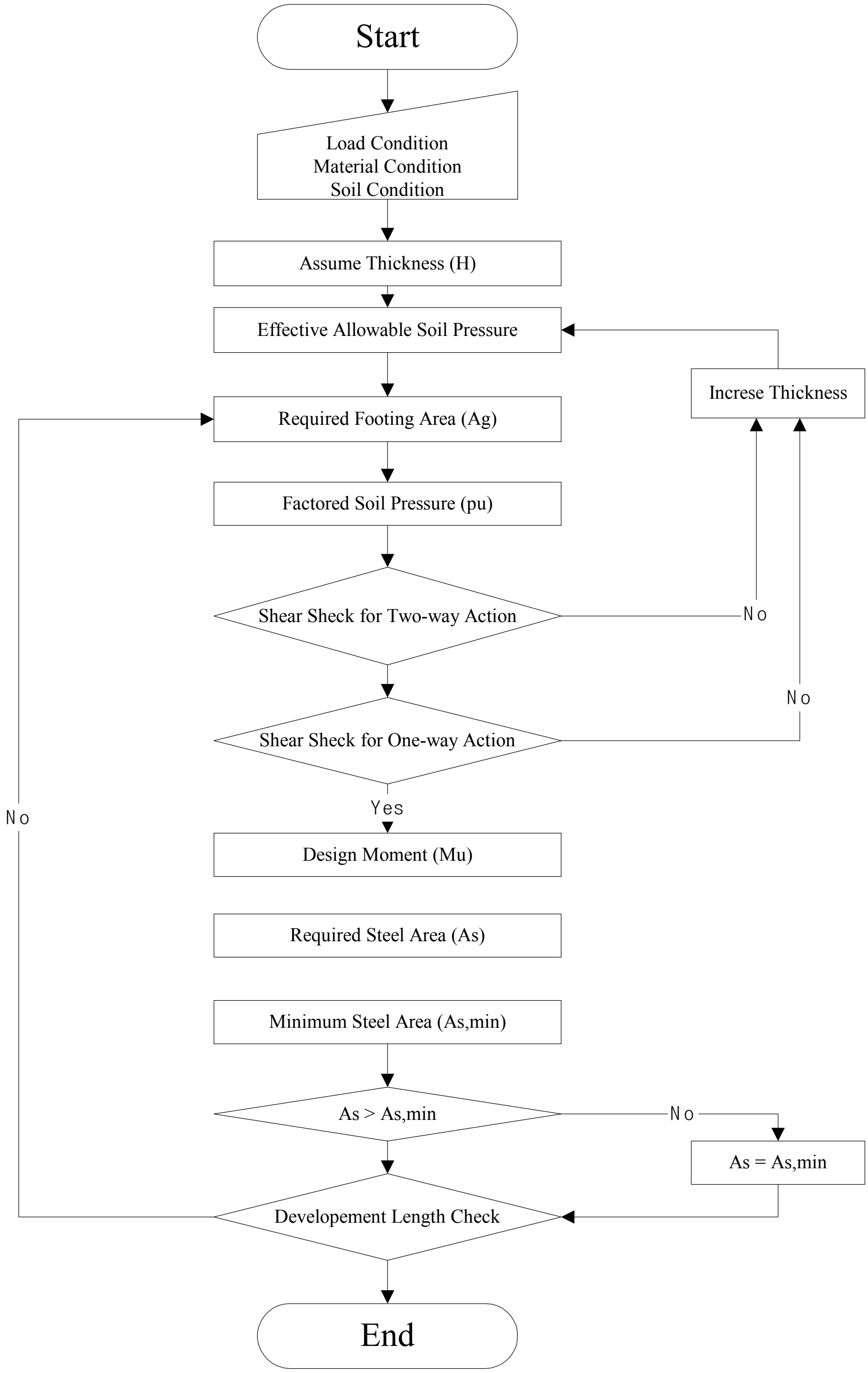
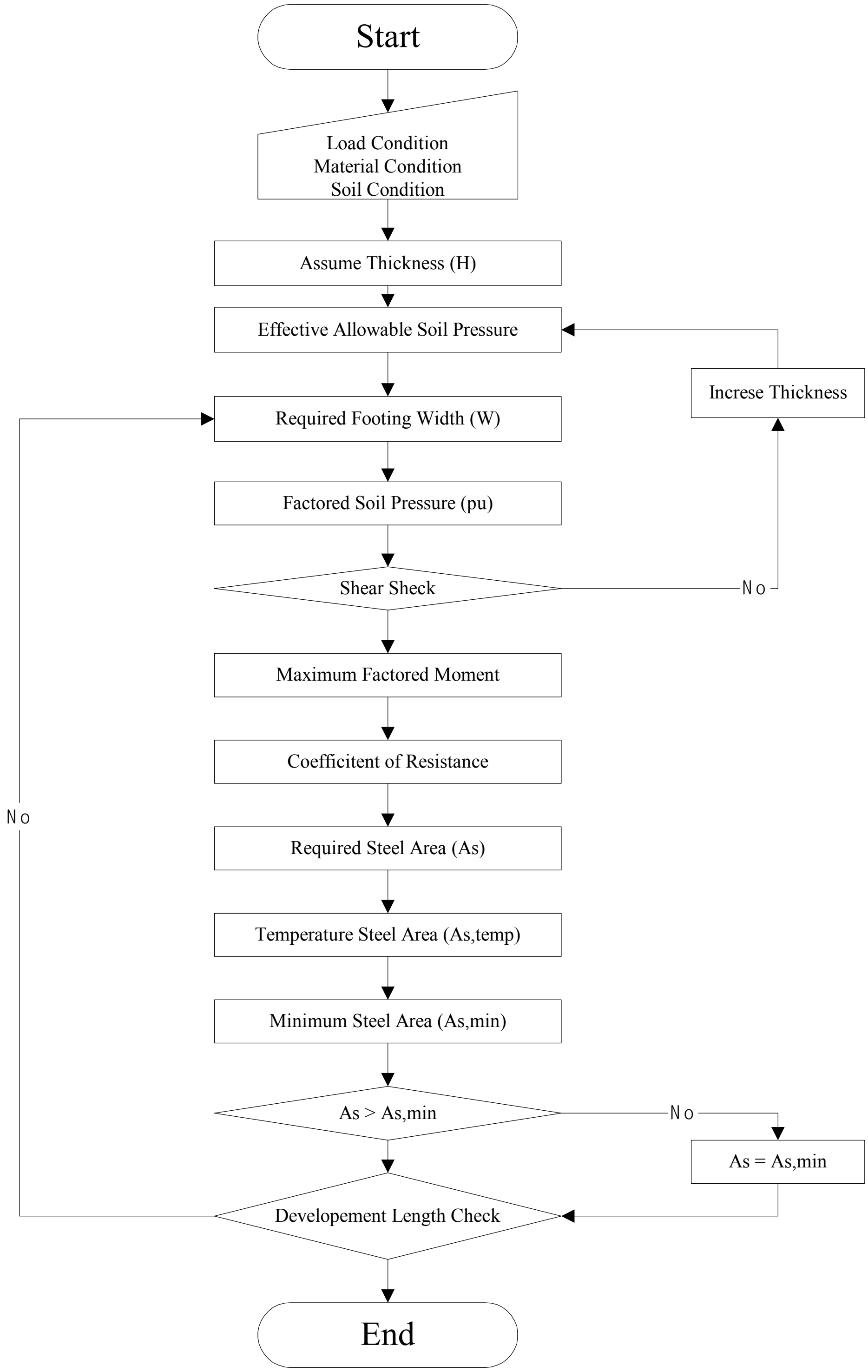



Fig. 9-3: Drawings of footing with 3D image

### 9.2 Flow Chart of Individual Column Footing



### *9.3 Flow Chart of Wall Footing*



## 10. Conclusion

RCSD program targets the architecture students. The ultimate goal of this program is to assist students in the reinforced concrete structures design and guide them to design structurally safe buildings. ACI Code is the most common code of R.C structure design, but it is difficult to use for beginner users.

The main purpose of this program is to provide as much basic information to users. RCSD does not restrict user to use just one answer but provides many possibility of structural member design for a set of building condition. Thus each calculation was divided into several steps and the *ASSIST* button was provided to guide users to give warning in case of incorrect input, provide definitions of new terms, provide typical values used in calculation. Also, Graphic User Interface (GUI) was used to procide visual output instead of numercal one and follow same color coding pattern in RCSD. All modules use blue to depict good design, red for bad design, green for compression value, and purple for tension value.

Several improvements can be made to RCSD, most important of which could be the inclusion of 3D graphical output. Most architecture students are familiar with 3D-computer graphics such as Autodesk AutoCAD. If RCSD uses the 3d graphic output, it will be really helpful to students to understand the structure and connection between structural members. Another improvement could be adding more design modules. RCSD provides six modules: One-way solid slab, Two-way slab, Column, Individual column footing, Wall footing and beam. Adding other modules, such as biaxial column and shear wall design module, would be useful.

There has not been enough time to actually test this program with student's actual design and to get feed back and add more *ASSIST* buttons. Microsoft Visual Basic has a user-friendly interface so modules can be added easily. I am hoping that another student will improve RCSD and develop it to make it an easier and more useful program.

## Bibliography

ACI, *ACI/PCA Seminar - Learn Significant Changes to the ACI 318-02 Building Code* [online], Available from: http://www.aci-int.org/Seminars/SeminarDetails.asp [Accessed 3/10/2002]

ACI Committee 340. *Design of two-way slabs: in accordance with the strength design method of ACI 318-83*, American Concrete Institute, 1985.

ACI Committee 340. *Design handbook: in accordance with the strength design method of ACI 318-83*, American Concrete Institute, 1984.

Ambrose, James E. *Simplified Design of Concrete Structures*, 7th ed., John Wiley & Sons, Inc., 1997.

French, Samuel E. *Reinforced concrete technology*, Delmar Publishers, 1994.

Gamble, William L., and Park, Robert. *Reinforced concrete slabs*, John Wiley & Sons, Inc., 2000.

Hassoun, M. Nadim. *Structural Concrete: theory and design*, Addison-Wesley Publishing Company, Inc., 1998.

Multiframe 4D Demo. Vers. 7.50. 10,978K. February 23, 2001. Daystar Software, Inc. [online], Available from: http://www.daystarsoftware.com/demo.html [Accessed 3/10/2002]

Nawy, Edward G. *Reinforced concrete: a fundamental approach*, Prentice Hall, 1996.

Nilson, Arthur H. *Design of Concrete Structures*, McGraw-Hill, Inc., 2000.

Parker, Harry, and Ambrose, James. *Simplified Design of Reinforced Concrete*, 5th ed., John Wiley & Sons, Inc., 1984.

PCA. *Notes on ACI 318-95 Building Code Requirements for Reinforced Concrete with Design Applications*, Portland Cement Association, 2000.

Prokon Calcpad Demo. Vers. 1.8. 4,120K. August 3, 2001. Prokon Software Consultants Ltd. [online], Available from: http://www.prokon.com/demo/dlstarted.htm [Accessed 2/14/2002]

Ray, S. S. *Reinforced Concrete: analysis and design*, Blackwell Science Ltd., 1994.

Rice, Paul F., and Hoffman, Edward S. *Structural Design Guide to the ACI Building Code*, 3rd ed., Van Nostrand Reinhold Company, Inc., 1985.

Spiegel, Leonard. Reinforced Concrete Design, 4th ed., Prentice-Hall, Inc., 1998.

Stephens, Lod. *Visual Basic Code Library*, John Wiley & Sons, Inc., 1999.

U.S. News, *Best Graduate School - Architecture Ranked in 1997* [online], Available from: http://www.usnews.com/usnews/edu/beyond/gradrank/gbarch.htm [Accessed 3/13/2002]

Williams, Alan. *Design of Reinforced Concrete Structures*, 2nd ed., Engineering press, 2000.

# Appendixes

## *1. Beam Module*

```
'<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
'  Reinforced Concrete Structure Designer (RCSD)
'<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
'  BEAM MODULE
'<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
'  University of Southern California, School of Architecture, Master of Building Science
'  Copyright 2002 by the University of Southern California and Kang-Kyu Choi
'  All right reserved
'  Contact Author: Kang-Kyu Choi  < hydrofall@hotmail.com >
'  RCSD is a reinforced concrete structure design assistant tool for learner.
'<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
Option Explicit
Dim BetaOne, requiredmo, maxsteelratio, depthstress, nominalmomentstrength As Double
Dim beamhv, beambv, TensionSteelArea, C_V, NC2, ESP, EY, Fsp, Asp, As2 As Double
Dim tarea, ylength, xlength, CaseBar, loadselectval As Double
Dim beamlocation, beamsupporttype, Reinforcementtype As Byte
Dim steelareaCaption, steeldiacaption, longbarspacing, drawcount, Centerx As Double
Dim steelareaCaption2, steeldiacaption2, longbarspacing2, drawcount2, Centerx2 As Double
Dim SAv, StirrupL, Comp_SP1, Comp_SP2, N3_SP, N4_SP, N5_SP As Double
Dim N_Value, N_Axis, I_CR, I_G, I_E, M_CR, ShortD_DL, ShortD_LL, Long_D, Long_Multi,
Long_R, Long_Ratio, Max_Steel, E_Thickness As Double
Dim MO_DL, MO_LL, D_AspR, Smax1, Smax2, Smax3, SMax4, STR_SIZE As Double

' <<<<<<<<<<<<<< Beam scaled drawing >>>>>>>>>>>>>>>>>
Private Sub beamb_Change()
If Val(beamh.Text) > Val(beamb.Text) Then
        Scale_Size.ScaleHeight = 1.5 * Val(beamh.Text)
        Scale_Size.ScaleWidth = 1.5 * Val(beamh.Text)
        Scale_Size.ScaleTop = -0.75 * Val(beamh.Text)
        Scale_Size.ScaleLeft = -0.75 * Val(beamh.Text)
        beamshape.Left = -0.5 * Val(beamb.Text)
        beamshape.Top = -0.5 * Val(beamh.Text)
        beamshape.Width = Val(beamb.Text)
        beamshape.Height = Val(beamh.Text)
Else
        Scale_Size.ScaleHeight = 1.5 * Val(beamb.Text)
        Scale_Size.ScaleWidth = 1.5 * Val(beamb.Text)
        Scale_Size.ScaleTop = -0.75 * Val(beamb.Text)
        Scale_Size.ScaleLeft = -0.75 * Val(beamb.Text)
        beamshape.Left = -0.5 * Val(beamb.Text)
        beamshape.Top = -0.5 * Val(beamh.Text)
        beamshape.Width = Val(beamb.Text)
        beamshape.Height = Val(beamh.Text)
End If
Slab_Shape.Top = Val(beamshape.Top) - Val(Slab_Shape.Height)
End Sub
Private Sub beamh_Change()
```

```
If Val(beamh.Text) > Val(beamb.Text) Then
            Scale_Size.ScaleHeight = 1.5 * Val(beamh.Text)
            Scale_Size.ScaleWidth = 1.5 * Val(beamh.Text)
            Scale_Size.ScaleTop = -0.75 * Val(beamh.Text)
            Scale_Size.ScaleLeft = -0.75 * Val(beamh.Text)
            beamshape.Left = -0.5 * Val(beamb.Text)
            beamshape.Top = -0.5 * Val(beamh.Text)
            beamshape.Width = Val(beamb.Text)
            beamshape.Height = Val(beamh.Text)
Else
            Scale_Size.ScaleHeight = 1.5 * Val(beamb.Text)
            Scale_Size.ScaleWidth = 1.5 * Val(beamb.Text)
            Scale_Size.ScaleTop = -0.75 * Val(beamb.Text)
            Scale_Size.ScaleLeft = -0.75 * Val(beamb.Text)
            beamshape.Left = -0.5 * Val(beamb.Text)
            beamshape.Top = -0.5 * Val(beamh.Text)
            beamshape.Width = Val(beamb.Text)
            beamshape.Height = Val(beamh.Text)
End If
Slab_Shape.Top = Val(beamshape.Top) - Val(Slab_Shape.Height)
End Sub

' <<<<<<<<<<<<<< select Beam support type >>>>>>>>>>>>>>>>>
Private Sub beamtypecombo_click()
Select Case beamtypecombo.ListIndex
            Case 0: beamsupporttype = 1
            Case 1: beamsupporttype = 2
            Case 2: beamsupporttype = 3
End Select
End Sub

' <<<<<<<<<<<<<< select Beam support type >>>>>>>>>>>>>>>>>
Private Sub Cant_select_Click()
beamtypecombo.Text = "CANTILEVER BEAM"
beamsupporttype = 3
End Sub

Private Sub ComstNum_Change()
If ComstNum.Text = 0 Then
               longbarpic.Cls
               ' Tension steel draw
               longbarspacing = (Val(beamb.Text) - (TenStNum.Text * steeldiacaption + 2)) /
            (Val(TenStNum.Text) - 0.9999)
               drawcount = 1
               Centerx = 0
               Do
                           longbarpic.Circle (Centerx, 0), steeldiacaption / 2, &H800080
                           drawcount = 1 + drawcount
                           Centerx = Centerx + longbarspacing + steeldiacaption
               Loop Until drawcount > Val(TenStNum.Text)
Else
               longbarpic.Cls
               ' Tension steel draw
               longbarspacing = (Val(beamb.Text) - (TenStNum.Text * steeldiacaption + 2))
            / (Val(TenStNum.Text) - 0.9999)
```

```
        drawcount = 1
        Centerx = 0
        Do
            longbarpic.Circle (Centerx, 0), steeldiacaption / 2, &H800080
            drawcount = 1 + drawcount
            Centerx = Centerx + longbarspacing + steeldiacaption
        Loop Until drawcount > Val(TenStNum.Text)

 'Compession steel draw
CaseBar = ComSTSize.ListIndex
Select Case CaseBar
    Case 0:
        steelareaCaption2 = 0.11
        steeldiacaption2 = 0.375
    Case 1:
        steelareaCaption2 = 0.2
        steeldiacaption2 = 0.5
    Case 2:
        steelareaCaption2 = 0.31
        steeldiacaption2 = 0.625
    Case 3:
        steelareaCaption2 = 0.44
        steeldiacaption2 = 0.75
    Case 4:
        steelareaCaption2 = 0.6
        steeldiacaption2 = 0.875
    Case 5:
        steelareaCaption2 = 0.79
        steeldiacaption2 = 1
    Case 6:
        steelareaCaption2 = 1
        steeldiacaption2 = 1.128
    Case 7:
        steelareaCaption2 = 1.27
        steeldiacaption2 = 1.27
    Case 8:
        steelareaCaption2 = 1.56
        steeldiacaption2 = 1.41
    Case 9:
        steelareaCaption2 = 2.25
        steeldiacaption2 = 1.693
    Case 10:
        steelareaCaption2 = 4
        steeldiacaption2 = 2.257
End Select
COMstTotal.Text = Val(steelareaCaption2) * Val(ComstNum.Text)
longbarspacing2 = (Val(beamb.Text) - (ComstNum.Text * steeldiacaption2 + 2)) / (Val(ComstNum.Text) - 0.9999)
drawcount2 = 1
Centerx2 = 0
Do
    longbarpic.Circle (Centerx2 - (Val(beamb.Text) / 100), -(Val(beamh.Text) * 8 / 10)), steeldiacaption2 / 2, &H4000&
    drawcount2 = 1 + drawcount2
    Centerx2 = Centerx2 + longbarspacing2 + steeldiacaption2
```

```
Loop Until drawcount2 > Val(ComstNum.Text)
End If
box.Left = -(steeldiacaption2 / 2 + 1)
box.Top = -beamh.Text + (steeldiacaption2 / 2 + 1)
box.Width = Val(beamb.Text)
box.Height = Val(beamh.Text)
End Sub

Private Sub ComSTSize_Click()
longbarpic.Cls
CaseBar = ComSTSize.ListIndex
Select Case CaseBar
        Case 0:
            steelareaCaption2 = 0.11
            steeldiacaption2 = 0.375
        Case 1:
            steelareaCaption2 = 0.2
            steeldiacaption2 = 0.5
        Case 2:
            steelareaCaption2 = 0.31
            steeldiacaption2 = 0.625
        Case 3:
            steelareaCaption2 = 0.44
            steeldiacaption2 = 0.75
        Case 4:
            steelareaCaption2 = 0.6
            steeldiacaption2 = 0.875
        Case 5:
            steelareaCaption2 = 0.79
            steeldiacaption2 = 1
        Case 6:
            steelareaCaption2 = 1
            steeldiacaption2 = 1.128
        Case 7:
            steelareaCaption2 = 1.27
            steeldiacaption2 = 1.27
        Case 8:
            steelareaCaption2 = 1.56
            steeldiacaption2 = 1.41
        Case 9:
            steelareaCaption2 = 2.25
            steeldiacaption2 = 1.693
        Case 10:
            steelareaCaption2 = 4
            steeldiacaption2 = 2.257
End Select
COMstTotal.Text = Val(steelareaCaption2) * Val(ComstNum.Text)
longbarspacing2 = (Val(beamb.Text) - (ComstNum.Text * steeldiacaption2 + 2)) / (Val(ComstNum.Text) - 0.9999)
drawcount2 = 1
Centerx2 = 0
```

```
Do
        longbarpic.Circle (Centerx2 - (Val(beamb.Text) / 100), -(Val(beamh.Text) * 8 / 10)), steeldiacaption2 / 2, &H4000&
        drawcount2 = 1 + drawcount2
        Centerx2 = Centerx2 + longbarspacing2 + steeldiacaption2
Loop Until drawcount2 > Val(ComstNum.Text)
longbarspacing = (Val(beamb.Text) - (TenStNum.Text * steeldiacaption + 2)) / (Val(TenStNum.Text) - 0.9999)
drawcount = 1
Centerx = 0
Do
        longbarpic.Circle (Centerx, 0), steeldiacaption / 2, &H800080
        drawcount = 1 + drawcount
        Centerx = Centerx + longbarspacing + steeldiacaption
Loop Until drawcount > Val(TenStNum.Text)
box.Left = -(steeldiacaption2 / 2 + 1)
box.Top = -beamh.Text + (steeldiacaption2 / 2 + 1)
box.Width = Val(beamb.Text)
box.Height = Val(beamh.Text)
End Sub

Private Sub D_BeamB_Change()
Pic_B.Text = D_BeamB.Text
End Sub

Private Sub D_BeamH_Change()
Pic_H.Text = D_BeamH.Text
End Sub

Private Sub D_MDL_Change()
D_MTL.Text = Val(D_MDL.Text) + Val(D_MLL.Text)
End Sub

Private Sub D_MLL_Change()
D_MTL.Text = Val(D_MDL.Text) + Val(D_MLL.Text)
End Sub

Private Sub Design_SPacing_Change()
PIC_SPACING.Text = Design_SPacing.Text
End Sub

Private Sub Dload_assist_Click()
SuperimposedDL.Show
End Sub

Private Sub Dmoment_assist_Click()
   MsgBox "Calculation of Design Moment is based on Factored Load," + vbCrLf + "NOT Service Load", , "Information"
End Sub

' <<<<<<<<<<<<<< select Beam position >>>>>>>>>>>>>>>>>
Private Sub ex_select_Click()
int_box.Visible = False
ex_box.Visible = True
beamlocation = 2
```

```
End Sub

' <<<<<<<<<<<<<<< select Beam support type >>>>>>>>>>>>>>>>>>
Private Sub Fix_select_Click()
beamtypecombo.Text = "FIXED BEAM"
beamsupporttype = 2
End Sub



' <<<<<<<<<<<<<<< select Beam position >>>>>>>>>>>>>>>>>>
Private Sub int_select_Click()
int_box.Visible = True
ex_box.Visible = False
beamlocation = 1
End Sub

Private Sub Lload_assist_Click()
load_select.Visible = True
occuassist.Visible = True
End Sub

Private Sub load_select_Click()
loadselectval = load_select.ListIndex
Select Case loadselectval
        Case 0: Beam_ll.Text = 150
        Case 1: Beam_ll.Text = 100
        Case 2: Beam_ll.Text = 100
        Case 3: Beam_ll.Text = 50
        Case 4: Beam_ll.Text = 40
        Case 5: Beam_ll.Text = 150
        Case 6: Beam_ll.Text = 150
        Case 7: Beam_ll.Text = 50
        Case 8: Beam_ll.Text = 40
        Case 9: Beam_ll.Text = 40
        Case 10: Beam_ll.Text = 250
        Case 11: Beam_ll.Text = 100
        Case 12: Beam_ll.Text = 125
        Case 13: Beam_ll.Text = 125
End Select
End Sub

Private Sub LONG_ALD_Change()
LONG_TLD.Text = Val(SHORT_TLD.Text) + Val(LONG_ALD.Text)
End Sub

Private Sub Material_assist_Click()
MsgBox " Typical Steel strength = 60000 psi " + vbCrLf + " Typical Conc. Strength = 4000 or 3000
psi ", vbInformation, "Typical Material Strength"
End Sub

Private Sub REBAR_ICON_Click()
Beam_PIC1.Visible = False
Beam_Pic2.Visible = True
End Sub
```

```
Private Sub ReturnSteel_Click()
beamtab.Tab = 4
End Sub

Private Sub ReturnThick3_Click()
beamtab.Tab = 1
End Sub


' <<<<<<<<<<<<<< select Beam support type >>>>>>>>>>>>>>>>>
Private Sub Simple_select_Click()
beamtypecombo.Text = "SIMPLE BEAM"
beamsupporttype = 1
End Sub

Private Sub Size_ICon_Click()
Beam_PIC1.Visible = True
Beam_Pic2.Visible = False
End Sub

Private Sub step_01_Click()
If beamlocation = "" Then
        beamlocation = 1
End If
If beamsupporttype = "" Then
        beamsupporttype = 1
End If

' <<<<<<<<<<<<<<< select dimentions according to the position >>>>>>>>>>>>>
If beamlocation = 1 Then
        xlength = Val(xlength1.Text)
        ylength = Val(ylength1.Text)
        tarea = Val(tarea1.Text)
End If
If beamlocation = 2 Then
        xlength = Val(xlength2.Text)
        ylength = Val(ylength2.Text)
        tarea = Val(tarea2.Text)
End If

'<<<<<<<<<<<<<<< compute the beam thickness >>>>>>>>>>>>>>>>>>>>>>>>>>

If beamsupporttype = 1 Then
        beamhv = 12 * (ylength / 16)
End If
If beamsupporttype = 2 Then
        beamhv = 12 * (ylength / 21)
End If
If beamsupporttype = 3 Then
        beamhv = 12 * (ylength / 8)
End If
beamh.Text = Round(beamhv)
beamb.Text = Round(0.5 * beamhv)
```

```
'<<<<<<<<<<<<<<<<<<<<< Scaled drawing procedure >>>>>>>>>>>>>>>>>>>>>>>>>
Scale_Size.ScaleHeight = 1.5 * beamhv
Scale_Size.ScaleWidth = 1.5 * beamhv
Scale_Size.ScaleTop = -0.75 * Val(beamh.Text)
Scale_Size.ScaleLeft = -0.75 * Val(beamh.Text)
beamshape.Width = Val(beamb.Text)
beamshape.Height = Val(beamh.Text)
beamshape.Left = -0.5 * Val(beamb.Text)
beamshape.Top = -0.5 * Val(beamh.Text)
Slab_Shape.Top = Val(beamshape.Top) - Val(Slab_Shape.Height)
beamtab.Tab = 1
End Sub

Private Sub step_02_Click()
Dim totaldl, totalll As Double
If beamh.Text < Round(beamhv) Then
	MsgBox "Beam thickness is lower than ACI recommendation !! " + vbCrLf + "This beam
needs to be checked for DEFLECTION!", , "Information"
End If

' Compute the load condition
uniformDL.Text = (Val(Beam_dl.Text) * tarea / ylength) / 1000
UniformDLSW.Text = Round((beamb.Text * Val(beamh.Text) * ylength * 150 / 144) / ylength) /
1000
UniformTDL.Text = Val(uniformDL.Text) + Val(UniformDLSW.Text)
UniformLL.Text = (Val(Beam_ll.Text) * tarea / ylength) / 1000
UniformFactorL.Text = 1.4 * UniformTDL.Text + 1.7 * UniformLL.Text

' Compute the required moment
If beamsupporttype = 1 Then
	requiredmo = Val(UniformFactorL.Text) * ylength * ylength / 8
End If
If beamsupporttype = 2 Then
	requiredmo = Val(UniformFactorL.Text) * ylength * ylength / 12
End If
If beamsupporttype = 3 Then
	'requiredMo = Val(UniformFactorL.Text) * ylength / 8
End If
designMO.Text = Fix(requiredmo * 100) / 100

'<<<<<<<<<< Compute the maximum possible moment >>>>>>>>>>>>>>>>
'<<<< maximum tension steel ratio >>>>
BetaOne = 0.85 - 0.05 * ((Val(Conc_type.Text) - 4000) / 1000)
If BetaOne > 0.85 Then
	BetaOne = 0.85
End If
If BetaOne < 0.65 Then
	BetaOne = 0.65
End If
maxsteelratio = (0.75 * 0.85 * BetaOne * (Conc_type.Text / Steel_type.Text) * (87000 / (87000 +
Steel_type.Text)))
MaxSTR.Text = Fix(maxsteelratio * 1000) / 1000

'<<<< maximum tension steel area calculation >>>>
TensionSteelArea = maxsteelratio * beamb.Text * (Val(beamh.Text) - 2.5)
```

```
TensSTArea.Text = Fix(TensionSteelArea * 100) / 100

'<<<< depth of the stress block >>>>
depthstress = TensionSteelArea * Val(Steel_type.Text) / (0.85 * Conc_type.Text * Val(beamb.Text))

'<<<< nominal moment strength >>>>
nominalmomentstrength = TensionSteelArea * Steel_type.Text * ((Val(beamh.Text) - 2.5) - (depthstress / 2)) / 12000
NominalMO.Text = Fix(nominalmomentstrength * 100) / 100

'<<<< Maximum possible Moment >>>>
MaxMO.Text = Fix((0.9 * NominalMO.Text) * 100) / 100

' Decide Beam reinforcement type according to the comparsion design and possible max moment
If Val(designMO.Text) < Val(MaxMO.Text) Then
	SingleDouble.Caption = "Design Moment < Max. Moment, SINGLE REINFORCEMENT NEEDED"
	Single_Shape.Visible = True
	Double_Shape.Visible = False
	COMST01.Visible = False
	ComST_TXT.Visible = False
	Reinforcementtype = 1
Else
	SingleDouble.Caption = "Design Moment > Max. Moment, DOUBLE REINFORCEMENT NEEDED"
	Single_Shape.Visible = False
	Double_Shape.Visible = True
	COMST01.Visible = True
	ComST_TXT.Visible = True
	Reinforcementtype = 2
End If
 beamtab.Tab = 2
 End Sub

Private Sub step_03_Click()
' 90% design tension steel
P_TenSTR.Text = 0.9 * (Val(MaxSTR.Text))
P_TenSTA.Text = Fix(0.9 * (Val(TensSTArea.Text)) * 100) / 100

' The moment capacity concrete - steel couple
'<<<< 90% depth of the stress block >>>>
depthstress = Val(P_TenSTA.Text) * Val(Steel_type.Text) / (0.85 * Conc_type.Text * Val(beamb.Text))

'<<<< 90% nominal moment strength >>>>
nominalmomentstrength = Val(P_TenSTA.Text) * Steel_type.Text * ((Val(beamh.Text) - 2.5) - (depthstress / 2)) / 12000

'<<<< 90% design Moment >>>>
P_TenMO.Text = Fix((0.9 * nominalmomentstrength) * 100) / 100

' Total required moment
R_TMO.Text = Val(designMO.Text)
' required compression moment
If Val(R_TMO.Text) > Val(P_TenMO.Text) Then
```

```
        R_COMMO.Text = Val(R_TMO.Text) - Val(P_TenMO.Text)
Else
        R_COMMO.Text = 0
End If

' Compute the required compressive force in the steel ( assume d'=2.5in. )
NC2 = (Val(R_COMMO.Text) * 12) / (0.9 * (Val(beamh.Text) - 2.5 - 2.5))

' Check the strain
' 90%]  depth of the stress block
depthstress = (Val(P_TenSTA.Text) * Val(Steel_type.Text)) / (0.85 * Conc_type.Text * Val(beamb.Text))
C_V = depthstress / BetaOne

' the unit strain at the centroid of the compression steel
ESP = 0.003 * (C_V - 2.5) / C_V

' strain Ey by table
If Val(Steel_type.Text) = 60000 Then
        EY = 0.00207
Else
        EY = 0.00138
End If

' Compressive steel yield
If ESP >= EY Then
        Fsp = Val(Steel_type.Text)
Else
        Fsp = ESP * 29000000
End If

' Compute As'
Asp = NC2 / (Fsp / 1000)
As2 = Fsp * Asp / (Val(Steel_type.Text))
R_ComSTA.Text = Fix(As2 * 100) / 100

' Total steel area As and As'
ToT_TSA.Text = Fix((Val(P_TenSTA.Text) + As2) * 100) / 100
ToT_CSA.Text = Fix(As2 * 100) / 100
beamtab.Tab = 3
End Sub

Private Sub Step_04_Click()
If Val(beamh.Text) > Val(beamb.Text) Then
        longbarpic.ScaleHeight = 1.5 * Val(beamh.Text)
        longbarpic.ScaleWidth = 1.5 * Val(beamh.Text)
Else
        longbarpic.ScaleHeight = 1.5 * Val(beamb.Text)
        longbarpic.ScaleWidth = 1.5 * Val(beamb.Text)
End If


If Val(beamh.Text) > Val(beamb.Text) Then
        longbarpic.ScaleTop = -0.75 * Val(longbarpic.ScaleHeight)
```

```
        longbarpic.ScaleLeft = -0.5 * (1.5 * Val(beamh.Text) - (Val(beamb.Text) - 4))
Else
        longbarpic.ScaleTop = -0.75 * Val(longbarpic.ScaleHeight)
        longbarpic.ScaleLeft = -0.2 * (1.5 * Val(beamb.Text))
End If

box.Left = -(steeldiacaption / 2 + 1)
box.Top = -beamh.Text + (steeldiacaption / 2 + 1)
box.Width = Val(beamb.Text)
box.Height = Val(beamh.Text)
longbarpic.Cls
beamtab.Tab = 4
End Sub

Private Sub Step_05_Click()
If Val(TenStTotal.Text) <= 0 Then
        MsgBox "Steel Area should be greater than ZERO !! ", , "Information"
End If

' Compute shear force
Vu_Max.Text = Val(UniformFactorL.Text) * ylength / 2
Vu_CR = Fix((Vu_Max - (Val(UniformFactorL.Text) * ((Val(beamh.Text) - 2.5) / 12))) * 100) / 100
V_C = Fix((0.85 * 2 * Sqr(Val(Conc_type.Text)) * Val(beamb.Text) * (Val(beamh.Text) - 2.5) / 
1000) * 100) / 100
half_VC.Text = (0.5 * V_C)

If Val(Vu_CR.Text) > Val(half_VC.Text) Then
        Stirrup_TXT.Caption = "Vu(max)<0.5øVc Stirrups are required."
        Stirrup_TXT.ForeColor = &HFF&
Else
        Stirrup_TXT.Caption = "Vu(max)>0.5øVc Stirrups are not required."
        Stirrup_TXT.ForeColor = &HFF0000
End If

' stirrup required length
StirrupL = (Val(Vu_Max.Text) - (0.5 * V_C)) / Val(UniformFactorL.Text)
STirrup_Length.Text = Fix(StirrupL * 100) / 100

' stirrup spacing and area of stirrup ratio
SAv = 0.85 * (Val(Steel_type.Text) / 1000) * (Val(beamh.Text) - 2.5) / (Vu_CR - V_C)

' the case of Vu on the critical section is smaller than Vc but bigger than Vc/2
N3_SP = 0.11 * SAv
N3_PSP.Text = Fix(N3_SP)
N4_SP = 0.2 * SAv
N4_PSP.Text = Fix(N4_SP)

' decide the minimum stirrup spacing
Comp_SP1 = 4 * Sqr(Conc_type.Text) / 1000 * beamb.Text * (Val(beamh.Text) - 2.5)
Comp_SP2 = (Vu_CR - V_C) / 0.85
Smax1 = (Val(beamh.Text) - 2.5) / 2


If Comp_SP1 >= Comp_SP2 Then
        Smax2 = 24
```

```
Else
        Smax2 = 12
End If

Smax3 = 0.11 * Val(Steel_type.Text) / (50 * Val(beamb.Text))
SMax4 = 0.2 * Val(Steel_type.Text) / (50 * Val(beamb.Text))

If Smax1 < Smax2 Then
        If Smax1 < Smax3 Then
                N3_MSP.Text = Smax1
        Else
                N3_MSP.Text = Smax3
        End If
Else
        If Smax2 < Smax3 Then
                N3_MSP.Text = Smax2
        Else
                N3_MSP.Text = Smax3
        End If
End If

If Smax1 < Smax2 Then
        If Smax1 < SMax4 Then
                N4_MSP.Text = Smax1
        Else
                N4_MSP.Text = SMax4
        End If
Else
        If Smax2 < SMax4 Then
                N4_MSP.Text = Smax2
        Else
                N4_MSP.Text = SMax4
        End If
End If

Design_Stirrup.Text = "# 3"

If Val(N3_PSP.Text) < Val(N3_MSP.Text) Then
        Design_SPacing.Text = N3_PSP.Text
Else
        Design_SPacing.Text = N3_MSP.Text
End If

beamtab.Tab = 5
End Sub

Private Sub Step_06_Click()
'<<<<<<<<<<<<<<<<<<< data transfer for deflection >>>>>>>>>>>>>>>>>>>
D_BeamB.Text = Val(beamb.Text)
D_BeamH.Text = Val(beamh.Text)
D_YLength = ylength


' Compute the Moment by Dead load and Live load
If beamsupporttype = 1 Then
```

```
        MO_DL = Val(UniformTDL.Text) * ylength * ylength / 8
        MO_LL = Val(UniformLL.Text) * ylength * ylength / 8
End If

If beamsupporttype = 2 Then
        MO_DL = Val(UniformTDL.Text) * ylength * ylength / 12
        MO_LL = Val(UniformLL.Text) * ylength * ylength / 12
End If

If beamsupporttype = 3 Then
        mo_dl = Val(UniformTDL.Text) * ylength / 8
        mo_ll = Val(UniformLL.Text) * ylength / 8
End If

D_MDL.Text = MO_DL
D_MLL.Text = MO_LL
D_MTL.Text = MO_DL + MO_LL
D_As.Text = Val(TenStTotal.Text)
D_ASP.Text = Val(COMstTotal.Text)
E_Thickness = Val(D_BeamH.Text) - 2.5

'<<<<<<<<<<<<<<< Deflection check >>>>>>>>>>>>>>>>>>>>>>>
' modular ratio n
If Val(Conc_type.Text) > 4000 Then
        If Val(Conc_type.Text) = 4000 Then
                N_Value = 8
        Else
                N_Value = 7
        End If
Else
        N_Value = 9
End If

' Neutral-axis location
 If Val(D_ASP.Text) > 0 Then
        N_Axis = N_Value * (Val(D_As.Text) + Val(D_ASP.Text)) * ((Sqr(1 + (2 * (Val(D_BeamB.Text)) * (E_Thickness + 3) / (N_Value * (Val(D_As.Text) + Val(D_ASP.Text)))))) - 1) / (Val(D_BeamB.Text))
Else
        N_Axis = N_Value * Val(D_As.Text) * ((Sqr(1 + (2 * (Val(D_BeamB.Text)) * E_Thickness / (N_Value * Val(D_As.Text))))) - 1) / (Val(D_BeamB.Text))
End If

' The moment inertia of the cracked section
I_CR = ((Val(D_BeamB.Text)) * (N_Axis ^ 3) / 3) + ((N_Value * Val(D_As.Text)) * (E_Thickness - N_Axis) ^ 2) + ((N_Value * Val(D_ASP.Text)) * (N_Axis - 3) ^ 2)

'The moment inertia of the gross section
I_G = (D_BeamH.Text ^ 3) * (Val(D_BeamB.Text)) / 12

' the moment would initially crack the cross section
M_CR = 7.5 * Sqr(Conc_type.Text / 1000000) * I_G / (D_BeamH.Text * 12 / 2)

' The effective moment of inertia
I_E = ((M_CR / Val(D_MTL)) ^ 3 * I_G) + (1 - ((M_CR / Val(D_MTL.Text)) ^ 3)) * I_CR
```

```
' THE Immediate dead load deflection
ShortD_DL = (5 * Val(D_MDL.Text) * (Val(D_YLength.Text) ^ 2) * (1728)) / (48 * 57 * Sqr(Val(Conc_type.Text)) * I_E)
ShortD_LL = (D_MLL.Text / D_MDL.Text) * ShortD_DL

' The Longterm deflection multiplier (DL+ sustained LL)
D_AspR = Val(D_ASP.Text) / (Val(D_BeamB.Text) * (Val(D_BeamH.Text) - 2.5))
Long_Multi = 2 / (1 + 50 * D_AspR)

' The longterm deflection
Long_D = ((Val(D_MDL.Text) + (0.5 * Val(D_MLL.Text))) / Val(D_MDL.Text)) * ShortD_DL * Long_Multi
SHORT_DLD.Text = Fix(ShortD_DL * 100) / 100
SHORT_LLD.Text = Fix(ShortD_LL * 100) / 100
SHORT_TLD.Text = Val(SHORT_DLD.Text) + Val(SHORT_LLD.Text)
LONG_ALD.Text = Fix(Long_D * 100) / 100

' maximum allowable deflection
Max_Deflection.Text = Fix((Val(D_YLength.Text) * 12 / 240) * 100) / 100

If Val(Max_Deflection.Text) >= Val(SHORT_TLD.Text) Then
        If Val(Max_Deflection.Text) >= Val(LONG_TLD.Text) Then
                Deflection_Label.ForeColor = &HFF0000
                Deflection_Label.Caption = "                    Good !!!
        Design beam thickness is adequate for the deflection! . "
                ReturnThick3.Visible = False
                ReturnSteel.Visible = False
        Else
                Deflection_Label.ForeColor = &HFF&
                Deflection_Label.Caption = " NOT Good!!!!! Design beam thickness is NOT
        adequate for the deflection. Check beam thickness!!."
                ReturnThick3.Visible = True
                ReturnSteel.Visible = True
        End If
Else
        Deflection_Label.ForeColor = &HFF&
        Deflection_Label.Caption = " NOT Good!!!!! Design beam thickness is NOT adequate for
the deflection. Check beam thickness!!."
        ReturnThick3.Visible = True
        ReturnSteel.Visible = True
End If
beamtab.Tab = 6
End Sub

Private Sub Step_07_Click()
beamtab.Tab = 7
End Sub

Private Sub Step_08_Click()
beamtab.Tab = 8
End Sub

Private Sub STR_1_Click()
If STR_1.Value = True Then
```

```
        If N3_PSP > N3_MSP Then
                Design_Stirrup.Text = "# 3"
                Design_SPacing.Text = N3_PSP.Text
        Else
                Design_Stirrup.Text = "# 3"
                Design_SPacing.Text = N3_MSP.Text
        End If
Else
        If N4_PSP > N4_MSP Then
                Design_Stirrup.Text = "# 4"
                Design_SPacing.Text = N4_PSP.Text
        Else
                Design_Stirrup.Text = "# 4"
                Design_SPacing.Text = N4_MSP.Text
        End If
End If
End Sub

Private Sub STR_2_Click()
If STR_1.Value = True Then
        If N3_PSP > N3_MSP Then
                Design_Stirrup.Text = "# 3"
                Design_SPacing.Text = N3_PSP.Text
        Else
                Design_Stirrup.Text = "# 3"
                Design_SPacing.Text = N3_MSP.Text
        End If
Else
        If N4_PSP > N4_MSP Then
                Design_Stirrup.Text = "# 4"
                Design_SPacing.Text = N4_PSP.Text
        Else
                Design_Stirrup.Text = "# 4"
                Design_SPacing.Text = N4_MSP.Text
        End If
End If
End Sub

Private Sub TenStNum_Change()

longbarpic.Cls
CaseBar = TenStSize.ListIndex

Select Case CaseBar
        Case 0:
          steelareaCaption = 0.11
          steeldiacaption = 0.375
        Case 1:
          steelareaCaption = 0.2
          steeldiacaption = 0.5
        Case 2:
          steelareaCaption = 0.31
          steeldiacaption = 0.625
        Case 3:
          steelareaCaption = 0.44
```

```
            steeldiacaption = 0.75
        Case 4:
            steelareaCaption = 0.6
            steeldiacaption = 0.875
        Case 5:
            steelareaCaption = 0.79
            steeldiacaption = 1
        Case 6:
            steelareaCaption = 1
            steeldiacaption = 1.128
        Case 7:
            steelareaCaption = 1.27
            steeldiacaption = 1.27
        Case 8:
            steelareaCaption = 1.56
            steeldiacaption = 1.41
        Case 9:
            steelareaCaption = 2.25
            steeldiacaption = 1.693
        Case 10:
            steelareaCaption = 4
            steeldiacaption = 2.257
End Select

TenStTotal.Text = Val(steelareaCaption) * Val(TenStNum.Text)
longbarspacing = (Val(beamb.Text) - (TenStNum.Text * steeldiacaption + 2)) / (Val(TenStNum.Text)
- 0.9999)
drawcount = 1
Centerx = 0

Do
        longbarpic.Circle (Centerx, 0), steeldiacaption / 2, &H800080
        drawcount = 1 + drawcount
        Centerx = Centerx + longbarspacing + steeldiacaption
Loop Until drawcount > Val(TenStNum.Text)

If Val(ComstNum.Text) <> 0 Then
        ' compression steel draw
        COMstTotal.Text = Val(steelareaCaption2) * Val(ComstNum.Text)
        longbarspacing2 = (Val(beamb.Text) - (ComstNum.Text * steeldiacaption2 + 2)) /
(Val(ComstNum.Text) - 0.9999)
        drawcount2 = 1
        Centerx2 = 0
        Do
                longbarpic.Circle (Centerx2 - (Val(beamb.Text) / 100), -(Val(beamh.Text) * 8 /
        10)), steeldiacaption2 / 2, &H4000&
                drawcount2 = 1 + drawcount2
                Centerx2 = Centerx2 + longbarspacing2 + steeldiacaption2
        Loop Until drawcount2 > Val(ComstNum.Text)
End If

box.Left = -(steeldiacaption / 2 + 1)
box.Top = -beamh.Text + (steeldiacaption / 2 + 1)
box.Width = Val(beamb.Text)
box.Height = Val(beamh.Text)
```

```
End Sub

Private Sub TenStSize_click()
longbarpic.Cls

' Tension steel draw
CaseBar = TenStSize.ListIndex
Select Case CaseBar
            Case 0:
               steelareaCaption = 0.11
               steeldiacaption = 0.375
            Case 1:
               steelareaCaption = 0.2
               steeldiacaption = 0.5
            Case 2:
               steelareaCaption = 0.31
               steeldiacaption = 0.625
            Case 3:
               steelareaCaption = 0.44
               steeldiacaption = 0.75
            Case 4:
               steelareaCaption = 0.6
               steeldiacaption = 0.875
            Case 5:
               steelareaCaption = 0.79
               steeldiacaption = 1
            Case 6:
               steelareaCaption = 1
               steeldiacaption = 1.128
            Case 7:
               steelareaCaption = 1.27
               steeldiacaption = 1.27
            Case 8:
               steelareaCaption = 1.56
               steeldiacaption = 1.41
            Case 9:
               steelareaCaption = 2.25
               steeldiacaption = 1.693
            Case 10:
               steelareaCaption = 4
               steeldiacaption = 2.257
End Select

TenStTotal.Text = Val(steelareaCaption) * Val(TenStNum.Text)
longbarspacing = (Val(beamb.Text) - (TenStNum.Text * steeldiacaption + 2)) / (Val(TenStNum.Text)
- 0.9999)
drawcount = 1
Centerx = 0
Do
            longbarpic.Circle (Centerx, 0), steeldiacaption / 2, &H800080
            drawcount = 1 + drawcount
            Centerx = Centerx + longbarspacing + steeldiacaption
Loop Until drawcount > Val(TenStNum.Text)
If Val(ComstNum.Text) <> 0 Then
            ' Compression steel draw
```

```
            COMstTotal.Text = Val(steelareaCaption2) * Val(ComstNum.Text)
            longbarspacing2 = (Val(beamb.Text) - (ComstNum.Text * steeldiacaption2 + 2)) / (Val(ComstNum.Text) - 0.9999)
            drawcount2 = 1
            Centerx2 = 0
            Do
                        longbarpic.Circle (Centerx2 - (Val(beamb.Text) / 100), -(Val(beamh.Text) * 8 / 10)), steeldiacaption2 / 2, &H4000&
                        drawcount2 = 1 + drawcount2
                        Centerx2 = Centerx2 + longbarspacing2 + steeldiacaption2
            Loop Until drawcount2 > Val(ComstNum.Text)
End If

box.Left = -(steeldiacaption / 2 + 1)
box.Top = -beamh.Text + (steeldiacaption / 2 + 1)
box.Width = Val(beamb.Text)
box.Height = Val(beamh.Text)
End Sub

Private Sub ToT_CSA_Change()
ComST_TXT.Caption = ToT_CSA.Text
Req_Com_ST.Text = ToT_CSA.Text
End Sub

Private Sub ToT_TSA_Change()
TenST_TXT.Caption = ToT_TSA.Text
Req_Ten_ST.Text = ToT_TSA.Text
End Sub

Private Sub V_C_Change()
half_VC.Text = (0.5 * V_C)
End Sub

Private Sub Vu_CR_Change()
Vu_CR1.Text = Vu_CR.Text
End Sub

Private Sub Vu_CR1_Change()
If Val(Vu_CR1.Text) > Val(half_VC.Text) Then
            Stirrup_TXT.Caption = "Vu > ½ΦVc, Stirrups are required."
            Stirrup_TXT.ForeColor = &HFF&
Else
            Stirrup_TXT.Caption = "Vu < ½ΦVc, Stirrups are NOT required."
            Stirrup_TXT.ForeColor = &HFF0000
End If

End Sub

Private Sub xlength1_Change()
tarea1.Text = Val(xlength1.Text) * Val(ylength1.Text)
End Sub


Private Sub xlength2_Change()
tarea2.Text = Val(xlength2.Text) * Val(ylength2.Text)
```

```
End Sub

Private Sub ylength1_Change()
tarea1.Text = Val(xlength1.Text) * Val(ylength1.Text)
End Sub

Private Sub ylength2_Change()
tarea2.Text = Val(xlength2.Text) * Val(ylength2.Text)
End Sub

Private Sub zzz_Click()
E_Thickness = Val(D_BeamH.Text) - 2.5
'<<<<<<<<<<<<<<< Deflection check >>>>>>>>>>>>>>>>>>>>>>
' modular ratio n
If Val(Conc_type.Text) > 4000 Then
        If Val(Conc_type.Text) = 4000 Then
                N_Value = 8
        Else
                N_Value = 7
        End If
Else
        N_Value = 9
End If

' Neutral-axis location
If Val(D_ASP.Text) > 0 Then
        N_Axis = N_Value * (Val(D_As.Text) + Val(D_ASP.Text)) * ((Sqr(1 + (2 * (Val(D_BeamB.Text)) * (E_Thickness + 2.5) / (N_Value * (Val(D_As.Text) + Val(D_ASP.Text)))))) - 1) / (Val(D_BeamB.Text))
Else
        N_Axis = N_Value * Val(D_As.Text) * ((Sqr(1 + (2 * (Val(D_BeamB.Text)) * E_Thickness / (N_Value * Val(D_As.Text))))) - 1) / (Val(D_BeamB.Text))
End If

' The moment inertia of the cracked section
I_CR = ((Val(D_BeamB.Text)) * (N_Axis ^ 3) / 3) + ((N_Value * Val(D_As.Text)) * (E_Thickness - N_Axis) ^ 2) + ((N_Value * Val(D_ASP.Text)) * (N_Axis - 2.5) ^ 2)

'The moment inertia of the gross section
I_G = (D_BeamH.Text ^ 3) * (Val(D_BeamB.Text)) / 12

' the moment would initially crack the cross section
M_CR = 7.5 * Sqr(Conc_type.Text / 1000000) * I_G / (D_BeamH.Text * 12 / 2)

' The effective moment of inertia
I_E = ((M_CR / Val(D_MTL)) ^ 3 * I_G) + (1 - ((M_CR / Val(D_MTL.Text)) ^ 3)) * I_CR

' THE Immediate dead load deflection
ShortD_DL = (5 * Val(D_MDL.Text) * (Val(D_YLength.Text) ^ 2) * (1728)) / (48 * 57 * Sqr(Val(Conc_type.Text)) * I_E)
ShortD_LL = (D_MLL.Text / D_MDL.Text) * ShortD_DL


' The Longterm deflection multiplier (DL+ sustained LL)
D_AspR = Val(D_ASP.Text) / (Val(D_BeamB.Text) * (Val(D_BeamH.Text) - 2.5))
```

```
Long_Multi = 2 / (1 + 50 * D_AspR)

' The longterm deflection
Long_D = ((Val(D_MDL.Text) + (0.5 * Val(D_MLL.Text))) / Val(D_MDL.Text)) * ShortD_DL *
Long_Multi
SHORT_DLD.Text = Fix(ShortD_DL * 100) / 100
SHORT_LLD.Text = Fix(ShortD_LL * 100) / 100
SHORT_TLD.Text = Val(SHORT_DLD.Text) + Val(SHORT_LLD.Text)
LONG_ALD.Text = Fix(Long_D * 100) / 100

' maximum allowable deflection
Max_Deflection.Text = Fix((Val(D_YLength.Text) * 12 / 240) * 100) / 100
If Val(Max_Deflection.Text) >= Val(SHORT_TLD.Text) Then
        If Val(Max_Deflection.Text) >= Val(LONG_TLD.Text) Then
                Deflection_Label.ForeColor = &HFF0000
                Deflection_Label.Caption = "                                        Good !!!
        Design beam thickness is adequate for the deflection! . "
                ReturnThick3.Visible = False
                ReturnSteel.Visible = False
        Else
                Deflection_Label.ForeColor = &HFF&
                Deflection_Label.Caption = " NOT Good!!!!! Design beam thickness is NOT
        adequate for the deflection. Check beam thickness!!."
                ReturnThick3.Visible = True
                ReturnSteel.Visible = True
        End If
  Else
        Deflection_Label.ForeColor = &HFF&
        Deflection_Label.Caption = " NOT Good!!!!! Design slab thickness is NOT adequate for the
deflection. Check beam thickness!!."
        ReturnThick3.Visible = True
        ReturnSteel.Visible = True
End If

End Sub
```

## *2. Column Module*

```
'<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
' Reinforced Concrete Structure Designer (RCSD)
'<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
' COLUMN MODULE
'<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
' University of Southern California, School of Architecture, Master of Building Science
' Copyright 2002 by the University of Southern California and Kang-Kyu Choi
' All right reserved
' Contact Author: Kang-Kyu Choi < hydrofall@hotmail.com >
' RCSD is a reinforced concrete structure design assistant tool for learner.
'<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
Option Explicit
Dim Slab_ALoad, Beam_ALoad, Column_ALoad, Total_ALoad, Applied_DL, Applied_LL,
Applied_TL, Tri_Area, LL_R As Double
Dim Slab_MLoad, Beam_MLoad, Total_MLoad, Applied_MDL, Applied_MLL, Applied_MO,
Reduction_F, E_Ratio, STR_Ratio  As Double
Dim Loadselectval, Col_Location, Connect_Type, Col_AreaOLD, Col_Area, Col_LengthX,
Col_LengthY, Load_Concrete, Load_Steel, Steel_Area, Steel_Ratio As Double
Dim Column_WidthOLD, Column_HeightOLD, Col_LengthXOLD, Col_WidthOLD As Double
Dim Casebar, Tiebar_Size, Casebar1, TieSpacing1, TieSpacing2, TieSpacing3 As Double
Dim PuAg, MuAgh, Alpha, Beta, Gamma, ST_Ratio As Double
Dim eh_V, PMX, PMY, Mainbar_size, Graph_Ratio As Double
Dim fy, fcp, b, h, dp, tas, cas, eccentricity, z1, z2, z3, c2, sc, nwval As Double
Dim fz, gz, hz, rz, sz, tz, uz, tabs, unew, vz, x1z As Double
Dim cubica, cubicb, cubicc, cubicd, Cvalue, pn, mn As Double
Dim EI, EIG, BetaD, Cm, Delta, P_Slender, NEW_Pu As Double

Private Sub Applied_Mu_Change()
If Val(Applied_Pu.Text) <= Val(Designed_Pu.Text) And Val(Applied_Mu.Text) <=
Val(Designed_Mu.Text) Then
        CHeck_Load.Caption = "Good !!"
        CHeck_Load.ForeColor = &HFF0000
        Return_COl.Visible = False
Else
        CHeck_Load.Caption = "Not Good !!"
        CHeck_Load.ForeColor = &HFF
        Return_COl.Visible = True
End If
End Sub

Private Sub Bar_ICon_Click()
Mainbar_Column.Visible = False
Cutted_Column.Visible = True
Size_Column.Visible = False
End Sub

Private Sub Barnum_Click()
Casebar = Barnum.ListIndex


Select Case Casebar
        Case 0:
```

```
            Mainbar_size = 0.11
        Case 1:
            Mainbar_size = 0.2
        Case 2:
            Mainbar_size = 0.31
        Case 3:
            Mainbar_size = 0.44
        Case 4:
            Mainbar_size = 0.6
        Case 5:
            Mainbar_size = 0.79
        Case 6:
            Mainbar_size = 1
        Case 7:
            Mainbar_size = 1.27
        Case 8:
            Mainbar_size = 1.56
        Case 9:
            Mainbar_size = 2.25
        Case 10:
            Mainbar_size = 4
End Select

Main_Quantity.Text = Round((STAREA1 / Mainbar_size) + 0.49)
BarSize_Info.Text = Barnum.Text
Design_Bar.Text = Mainbar_size * Val(Main_Quantity.Text)

If Mainbar_size < 1.27 Then
        Recommened_Tie.Text = "NO. 3"
Else
        Recommened_Tie.Text = "NO. 4"
End If
End Sub

Private Sub Column_Height_Change()
' for prevent type mismatch error by null inputing
ZeroD Column_Height
If Val(Column_Width.Text) >= Val(Column_Height.Text) Then
        Col_Drawbox.ScaleHeight = 1.5 * Column_Width.Text
        Col_Drawbox.ScaleWidth = 1.5 * Column_Width.Text
        Col_Drawbox.ScaleTop = -(0.75 * Column_Width.Text)
        Col_Drawbox.ScaleLeft = -(0.75 * Column_Width.Text)
        Column_shape.Height = Val(Column_Height.Text)
        Column_shape.Width = Val(Column_Width.Text)
        Column_shape.Top = -(Column_Height.Text / 2)
        Column_shape.Left = -(Column_Width.Text / 2)
Else
        Col_Drawbox.ScaleHeight = 1.5 * Column_Height.Text
        Col_Drawbox.ScaleWidth = 1.5 * Column_Height.Text
        Col_Drawbox.ScaleTop = -(0.75 * Column_Height.Text)
        Col_Drawbox.ScaleLeft = -(0.75 * Column_Height.Text)
        Column_shape.Height = Val(Column_Height.Text)
        Column_shape.Width = Val(Column_Width.Text)
        Column_shape.Top = -(Column_Height.Text / 2)
        Column_shape.Left = -(Column_Width.Text / 2)
```

```
End If

If Val(Factor_MomentX.Text) = 0 Then
'Steel Area Calculate
        Load_Concrete = 0.8 * 0.7 * 0.85 * (Conc_type.Text / 1000) * (Column_Width.Text * Column_Height.Text) * (1 - 0.03)
        Load_Steel = Val(Factor_Axial.Text) - Load_Concrete
        Steel_Area = Load_Steel / (0.8 * 0.7 * (Val(Steel_type.Text) / 1000))
Else
        PuAg = Factor_Axial.Text / (Column_Width.Text * Column_Height.Text)
        MuAgh = (Factor_MomentX.Text * 12) / ((Column_Width.Text * Column_Height.Text) * (Column_Width.Text))
        Gamma = 0.04 * (MuAgh ^ 2)
        Alpha = ((Val(Steel_type.Text / 1000) - 60) * 0.0062) + ((Val(Conc_type.Text / 1000) - 4) * 0.104)
        Beta = PuAg - 0.96 + Alpha
        ST_Ratio = (Beta + Sqr(Beta ^ 2 + (144 * Gamma))) / 72
        Steel_Area = ST_Ratio * (Val(Column_Width.Text) * Val(Column_Height.Text))
End If

' to prevent (-) rebar size
If Steel_Area < 0 Then
        Steel_Area = 0
End If

' Check the steel ratio
STArea.Text = Fix(Steel_Area * 100) / 100
Steel_Ratio = Fix((Steel_Area / (Column_Width.Text * Column_Height.Text)) * 100 * 100) / 100
STRatio.Text = Steel_Ratio

' Give information under-reinforced column and over reinforced column
If Steel_Ratio > 5 Then
        STRatio_INfo.Caption = "                    CAUTION!!                    Over reinforced column !!!                    Typical Steel Ratio is Greater than 2% and Less than 5%.                    Increase column size!"
        STRatio_INfo.ForeColor = &HFF&
Else
        If Steel_Ratio < 2 Then
                STRatio_INfo.Caption = "                    CAUTION!!                    Under reinforced column !!!                    Typical Steel Ratio is Greater than 2% and Less than 5%. Decrease column size!"
                STRatio_INfo.ForeColor = &HFF&
        Else
                STRatio_INfo.Caption = "                    GOOD!!                    Typical Steel Ratio is Greater than 2% and Less than 5%."
                STRatio_INfo.ForeColor = &HFF0000
        End If
End If

Column_Width1.Text = Column_Width.Text
Column_Height1.Text = Column_Height.Text
End Sub

Private Sub Column_Height_GotFocus()
selectfield Column_Height
```

```
End Sub

Private Sub Column_Height_KeyPress(KeyAscii As Integer)
Select Case KeyAscii
        Case 48 To 57, 8
        Case Else
        KeyAscii = 0
        End Select
End Sub

Private Sub Column_Width_Change()
' for prevent type mismatch error by null inputing
ZeroD Column_Width

If Val(Column_Width.Text) >= Val(Column_Height.Text) Then
        Col_Drawbox.ScaleHeight = 1.5 * Column_Width.Text
        Col_Drawbox.ScaleWidth = 1.5 * Column_Width.Text
        Col_Drawbox.ScaleTop = -(0.75 * Column_Width.Text)
        Col_Drawbox.ScaleLeft = -(0.75 * Column_Width.Text)
        Column_shape.Height = Val(Column_Height.Text)
        Column_shape.Width = Val(Column_Width.Text)
        Column_shape.Top = -(Column_Height.Text / 2)
        Column_shape.Left = -(Column_Width.Text / 2)
Else
        Col_Drawbox.ScaleHeight = 1.5 * Column_Height.Text
        Col_Drawbox.ScaleWidth = 1.5 * Column_Height.Text
        Col_Drawbox.ScaleTop = -(0.75 * Column_Height.Text)
        Col_Drawbox.ScaleLeft = -(0.75 * Column_Height.Text)
        Column_shape.Height = Val(Column_Height.Text)
        Column_shape.Width = Val(Column_Width.Text)
        Column_shape.Top = -(Column_Height.Text / 2)
        Column_shape.Left = -(Column_Width.Text / 2)
End If

If Val(Factor_MomentX.Text) = 0 Then
        'Steel Area Calculate
        Load_Concrete = 0.8 * 0.7 * 0.85 * (Conc_type.Text / 1000) * (Column_Width.Text *
Column_Height.Text) * (1 - 0.03)
        Load_Steel = Val(Factor_Axial.Text) - Load_Concrete
        Steel_Area = Load_Steel / (0.8 * 0.7 * (Val(Steel_type.Text) / 1000))
Else
        PuAg = Factor_Axial.Text / (Column_Width.Text * Column_Height.Text)
        MuAgh = (Factor_MomentX.Text * 12) / ((Column_Width.Text * Column_Height.Text) *
(Column_Width.Text))
        Gamma = 0.04 * (MuAgh ^ 2)
        Alpha = ((Val(Steel_type.Text / 1000) - 60) * 0.0062) + ((Val(Conc_type.Text / 1000) - 4) *
0.104)
        Beta = PuAg - 0.96 + Alpha
        ST_Ratio = (Beta + Sqr(Beta ^ 2 + (144 * Gamma))) / 72
        Steel_Area = ST_Ratio * (Val(Column_Width.Text) * Val(Column_Height.Text))
End If


' to prevent (-) rebar size
If Steel_Area < 0 Then
```

```
            Steel_Area = 0
End If

' Check the steel ratio
STArea.Text = Fix(Steel_Area * 100) / 100
Steel_Ratio = Fix((Steel_Area / (Column_Width.Text * Column_Height.Text)) * 100 * 100) / 100
STRatio.Text = Steel_Ratio

' Give information under-reinforced column and over reinforced column
If Steel_Ratio > 5 Then
            STRatio_INfo.Caption = "                    CAUTION!!                    Over reinforced column
!!!                    Typical Steel Ratio is Greater than 2% and Less than 5%.                    Increase
column size!"
            STRatio_INfo.ForeColor = &HFF&
Else
            If Steel_Ratio < 2 Then
                        STRatio_INfo.Caption = "                    CAUTION!!                    Under
reinforced column !!!                    Typical Steel Ratio is Greater than 2% and Less than 5%.
Decrease column size!"
                        STRatio_INfo.ForeColor = &HFF&
            Else
                        STRatio_INfo.Caption = "                    GOOD!!                    Typical Steel
Ratio is Greater than 2% and Less than 5%."
                        STRatio_INfo.ForeColor = &HFF0000
            End If
End If

Column_Width1.Text = Column_Width.Text
Column_Height1.Text = Column_Height.Text
End Sub

Private Sub Column_Width_GotFocus()
selectfield Column_Width
End Sub

Private Sub Column_Width_KeyPress(KeyAscii As Integer)
Select Case KeyAscii
            Case 48 To 57, 8
            Case Else
            KeyAscii = 0
            End Select
End Sub

Private Sub Connect_Beam_click()
Select Case Connect_Beam.ListIndex
Case 0: Connect_Type = 0
            Pic_MO.Visible = True
            Pic_SIM.Visible = False
Case 1: Connect_Type = 1
            Pic_MO.Visible = False
            Pic_SIM.Visible = True
End Select
End Sub
Private Sub Corner_select_Click()
Col_Location = 2
```

```
Column_mark.Left = 600
Column_mark.Top = 2150
Tri_Area0.Visible = False
Tri_Area1.Visible = False
Tri_Area2.Visible = True
Beam0.Visible = False
Beam1.Visible = False
Beam2.Visible = True
End Sub

Private Sub D_Load_gotfocus()
'selectfield D_Load
End Sub

Private Sub Designed_eh_Change()
If Val(Designed_eh.Text) > 0.1 Then
        eh_V = (2 / Val(Designed_eh.Text))
Else
        eh_V = 20
End If

eh_line.X2 = 2
eh_line.Y2 = -eh_V
End Sub

Private Sub Designed_Mu_Change()
If Val(Applied_Pu.Text) <= Val(Designed_Pu.Text) And Val(Applied_Mu.Text) <=
Val(Designed_Mu.Text) Then
        CHeck_Load.Caption = "Good !!"
        CHeck_Load.ForeColor = &HFF0000
        Return_COl.Visible = False
Else
        CHeck_Load.Caption = "Not Good !!"
        CHeck_Load.ForeColor = &HFF
        Return_COl.Visible = True
End If
End Sub

Private Sub Dload_assist_Click()
SuperimposedDLC.Show
End Sub

Private Sub Ext_select_Click()
Col_Location = 1
Column_mark.Left = 600
Column_mark.Top = 1050
Tri_Area0.Visible = False
Tri_Area1.Visible = True
Tri_Area2.Visible = False
Beam0.Visible = False
Beam1.Visible = True
Beam2.Visible = False
End Sub
Private Sub Factor_Axial_Change()
Applied_Pu.Text = Factor_Axial.Text
```

```
End Sub

Private Sub Factor_MomentX_Change()
NullD Factor_MomentX
Applied_Mu.Text = Factor_MomentX
E_Value.Text = Fix((Factor_MomentX.Text * 12 / Factor_Axial.Text) * 100) / 100
End Sub

Private Sub Int_select_Click()
Col_Location = 0
Column_mark.Left = 2750
Column_mark.Top = 1050
Tri_Area0.Visible = True
Tri_Area1.Visible = False
Tri_Area2.Visible = False
Beam0.Visible = True
Beam1.Visible = False
Beam2.Visible = False
End Sub

Private Sub load_assist_Click()
load_select.Visible = True
occuassist.Visible = True
End Sub

Private Sub Lload_assist_Click()
load_select.Visible = True
occuassist.Visible = True
End Sub

Private Sub load_select_click()
Loadselectval = load_select.ListIndex
Select Case Loadselectval
        Case 0: Column_ll.Text = 150
        Case 1: Column_ll.Text = 100
        Case 2: Column_ll.Text = 100
        Case 3: Column_ll.Text = 50
        Case 4: Column_ll.Text = 40
        Case 5: Column_ll.Text = 150
        Case 6: Column_ll.Text = 150
        Case 7: Column_ll.Text = 50
        Case 8: Column_ll.Text = 40
        Case 9: Column_ll.Text = 40
        Case 10: Column_ll.Text = 250
        Case 11: Column_ll.Text = 100
        Case 12: Column_ll.Text = 125
        Case 13: Column_ll.Text = 125
End Select
End Sub

Private Sub Main_Quantity_Change()
Design_Bar.Text = Mainbar_size * Val(Main_Quantity.Text)

' Check the quantities of bar
If Val(Main_Quantity.Text) < 4 Or Val(Main_Quantity.Text) > 14 Then
```

```
            Quantity_help.Visible = True
Else
            Quantity_help.Visible = False
End If

' Check the bar is the even or odd number
If InStr(1, (Val(Main_Quantity.Text) / 2), ".") <> 0 Then
            Quantity_help.Visible = True
Else
            Quantity_help.Visible = False
End If
End Sub

Private Sub Material_assist_Click()
MsgBox " Typical Steel strength = 60000 psi " + vbCrLf + " Typical Conc. Strength = 4000 or 3000
psi ", vbInformation, "Typical Material Strength"
End Sub

Private Sub Moment_Connect_Click()
Connect_Type = 0
Pic_MO.Visible = True
Pic_SIM.Visible = False
Connect_Beam.Text = "Moment Connection"
Corbel.Visible = False
End Sub

Private Sub Reinforce_Icon_Click()
Mainbar_Column.Visible = True
Cutted_Column.Visible = False
Size_Column.Visible = False
End Sub

Private Sub Return_COl_Click()
Column_Tab.Tab = 2
End Sub

Private Sub Return_slender_Click()
NEW_Pu = Val(Factor_MomentX.Text) * Delta
If Val(Factor_MomentX.Text) > 0 Then
            Factor_MomentX.Text = NEW_Pu
Else
            Factor_MomentX.Text = Val(Factor_Axial.Text) * ((0.6 + 0.03 * Val(Column_Width.Text))
/ 12) * Delta
End If

If Val(Column_Width.Text) < Val(E_Value.Text) Then
            Column_Width.Text = E_Value.Text / 0.8
End If

End Sub


Private Sub Simple_Connect_Click()
Connect_Type = 1
```

```
Pic_MO.Visible = False
Pic_SIM.Visible = True
Connect_Beam.Text = "Simply Supported"
Corbel.Visible = True
End Sub

Private Sub Size_Icon_Click()
Mainbar_Column.Visible = False
Cutted_Column.Visible = False
Size_Column.Visible = True
End Sub

Private Sub SpacingNum_Click()
Casebar1 = SpacingNum.ListIndex
Select Case Casebar1
        Case 0:
                Tiebar_Size = 0.11
        Case 1:
                Tiebar_Size = 0.2
        Case 2:
                Tiebar_Size = 0.31
End Select

TieSpacing1 = 48 * Sqr(4 * Tiebar_Size / 3.14)
TieSpacing2 = 16 * Sqr(4 * Mainbar_size / 3.14)

If Val(Column_Width.Text) > Val(Column_Height.Text) Then
        TieSpacing3 = Val(Column_Width.Text)
Else
        TieSpacing3 = Val(Column_Height.Text)
End If

If TieSpacing1 < TieSpacing2 Then
        If TieSpacing1 < TieSpacing3 Then
                Tie_Spacing.Text = Round(TieSpacing1)
        Else
                Tie_Spacing.Text = Round(TieSpacing3)
        End If
Else
        If TieSpacing2 < TieSpacing3 Then
                Tie_Spacing.Text = Round(TieSpacing2)
        Else
                Tie_Spacing.Text = Round(TieSpacing3)
        End If
End If

Tie_6BIG.Visible = True
End Sub

Private Sub STArea_Change()
ZeroD STArea
' Check the steel ratio
Steel_Ratio = Fix((STArea.Text / (Column_Width.Text * Column_Height.Text)) * 100 * 100) / 100
```

```
STRatio.Text = Steel_Ratio

' Give information under-reinforced column and over reinforced column
If Steel_Ratio > 5 Then
	STRatio_INfo.Caption = "                    CAUTION!!                  Over reinforced column !!!                    Typical Steel Ratio is Greater than 2% and Less than 5%.                    Increase column size!"
	STRatio_INfo.ForeColor = &HFF&
Else
	If Steel_Ratio < 2 Then
		STRatio_INfo.Caption = "                    CAUTION!!                    Under reinforced column !!!                    Typical Steel Ratio is Greater than 2% and Less than 5%. Decrease column size!"
		STRatio_INfo.ForeColor = &HFF&
	Else
		STRatio_INfo.Caption = "                    GOOD!!                    Typical Steel Ratio is Greater than 2% and Less than 5%."
		STRatio_INfo.ForeColor = &HFF0000
	End If
End If

Column_Width1.Text = Column_Width.Text
Column_Height1.Text = Column_Height.Text
STAREA1.Text = STArea.Text
End Sub

Private Sub STArea_GotFocus()
selectfield STArea
End Sub

Private Sub STArea_KeyPress(KeyAscii As Integer)
Select Case KeyAscii
	Case 48 To 57, 8
	Case Else
	KeyAscii = 0
End Select
End Sub

Private Sub step_01_Click()
Column_Tab.Tab = 1
End Sub

Private Sub ZeroD(ByVal Zero_Divide As TextBox)
If Zero_Divide.Text = "" Then
	Zero_Divide.Text = 1
	Zero_Divide.SelStart = 0
	Zero_Divide.SelLength = 5
End If
If Zero_Divide.Text = 0 Then
	Zero_Divide.Text = 1
	Zero_Divide.SelStart = 0
	Zero_Divide.SelLength = 5
End If
If Zero_Divide.Text > 999999 Then
```

```
            Zero_Divide.Text = 1
            Zero_Divide.SelStart = 0
            Zero_Divide.SelLength = 5
End If
End Sub

Private Sub selectfield(ByVal text_box As TextBox)
text_box.SelStart = 0
text_box.SelLength = Len(text_box.Text)
End Sub

Private Sub Step_02_Click()
' Tributary area calculate
Tri_Area = Val(Tri_Lx.Text) * Val(Tri_Ly.Text)

' Slab self weight calculate
Slab_ALoad = (Tri_Area * (Val(Slab_Thick.Text) / 12) * 150 * (Val(FL_Num.Text) + 1)) / 1000

' beam self weight calculate
Beam_ALoad = (((Val(BeamA_Width.Text) * Val(BeamA_height.Text) * Val(Tri_Lx.Text) * 150 /
144) + (Val(BeamB_Width.Text) * Val(BeamB_Height.Text) * Val(Tri_Ly.Text) * 150 / 144)) *
(Val(FL_Num.Text) + 1)) / 1000

' Dead load calculate
Applied_DL = (Tri_Area * Val(Column_dl.Text)) * (Val(FL_Num.Text) + 1) / 1000

' Live load reduction factor
LL_R = (((Tri_Area - 150) * (0.08)) / 100)
If LL_R > 0 Then
            If LL_R < 0.4 Then
                        LL_R = 1 - LL_R
            Else
                        LL_R = 0.6
            End If
Else
            LL_R = 1
End If

' Modified Live Load calculate
Applied_LL = LL_R * (Tri_Area * Val(Column_ll.Text)) * (Val(FL_Num.Text) + 1) / 1000

' compute the total load with out column self weight
Applied_TL = 1.4 * (Slab_ALoad + Beam_ALoad + Applied_DL) + 1.7 * Applied_LL

' Assume column size depend on the total load (W/o column)
If Applied_TL > 400 Then
   If Applied_TL < 1000 Then
            Column_ALoad = 1.5 * 1.5 * Val(FL_Height.Text) * 150 * (Val(FL_Num.Text)) / 1000
   Else
            Column_ALoad = 2 * 2 * Val(Val(FL_Height.Text)) * 150 * (Val(FL_Num.Text)) / 1000
   End If
Else
   Column_ALoad = 1 * 1 * Val(Val(FL_Height.Text)) * 150 * (Val(FL_Num.Text)) / 1000
End If
'Compute all Load
```

```
Applied_TL = Applied_TL + Column_ALoad
Factor_Axial.Text = Fix(Applied_TL)

If Col_Location = 1 Then
        Slab_MLoad = (Tri_Area * (Val(Slab_Thick.Text) / 12) * 150) / 1000
        Beam_MLoad = (Val(BeamA_Width.Text) * Val(BeamA_height.Text) * Val(Tri_Lx.Text)
* 150 / 144) / 1000
        Applied_MDL = (Tri_Area * Val(Column_dl.Text)) / 1000
        Applied_MLL = (Tri_Area * Val(Column_ll.Text)) / 1000
        Total_MLoad = Slab_MLoad + Beam_MLoad + Applied_MDL + Applied_MLL
Else
        Total_MLoad = 0
End If

If Connect_Type = 0 Then
        Applied_MO = Total_MLoad * Val(Tri_Lx.Text) / 2
Else
        Applied_MO = Total_MLoad * ((Val(Corbel_Sp.Text) / 12) + 1)
End If

Factor_MomentX.Text = Fix(Applied_MO)
E_Value.Text = Fix(Factor_MomentX.Text * 12 / Factor_Axial.Text)

' Assume Column Area with 3% Steel Reinforcement
Col_AreaOLD = Val(Factor_Axial.Text) / (0.8 * 0.7 * ((0.85 * (Conc_type.Text / 1000) * (1 - 0.03))
+ ((Steel_type.Text / 1000) * 0.03)))

If Val(E_Value.Text) > 1 Then
        Col_LengthXOLD = Fix(1.5 * Sqr(Col_AreaOLD / 1.5))
Else
        Col_LengthXOLD = Fix(Sqr(Col_AreaOLD))
End If

E_Ratio = Fix(Val(E_Value.Text) * 100 / (Col_LengthXOLD))

Select Case E_Ratio
        Case 0 To 2: Reduction_F = 1
        Case 3 To 10: Reduction_F = 0.85
        Case 11 To 20: Reduction_F = 0.75
        Case 21 To 30: Reduction_F = 0.65
        Case 31 To 40: Reduction_F = 0.55
        Case 41 To 50: Reduction_F = 0.5
        Case 51 To 60: Reduction_F = 0.425
        Case 61 To 70: Reduction_F = 0.375
        Case 71 To 80: Reduction_F = 0.325
        Case 81 To 90: Reduction_F = 0.3
        Case 91 To 100: Reduction_F = 0.27
End Select

' Redesign according to reduction factor.
Col_Area = Col_AreaOLD / Reduction_F


If Val(E_Value.Text) > 1 Then
```

```
            Col_LengthX = Fix(Sqr(Col_Area / 1.5))
            Column_Width.Text = 1.5 * Col_LengthX
            Column_Height.Text = Col_LengthX
Else
            Col_LengthX = Fix(Sqr(Col_AreaOLD))
            Column_Width.Text = Col_LengthX
            Column_Height.Text = Col_LengthX
End If

' Column Drawing
Column_Tab.Tab = 2
End Sub

Private Sub Step_03_Click()
Designed_eh.Text = Fix(E_Value.Text * 100 / Column_Width.Text) / 100
'<<<<<<<<<<<<<<<<<<<<<<<<< Slenderness  Check >>>>>>>>>>>>>>>>>>>>>>>>>>>>>
KLR.Text = Fix(0.65 * Val(FL_Height.Text) * 12 / (0.3 * Val(Column_Width.Text)))
If Val(Factor_MomentX.Text) > 0 Then
            MAX_KLR.Text = 34 + 12 * (Factor_MomentX.Text / Factor_MomentX.Text)
Else
            MAX_KLR.Text = 34
End If

If Val(KLR.Text) > Val(MAX_KLR.Text) Then
            EIG = 57 * Sqr(Val(Conc_type.Text)) * 1000 * Val(Column_Height.Text) *
Val(Column_Width.Text) ^ 3 / 12
            BetaD = (Applied_TL - 1.7 * Applied_LL) / Applied_TL
            EI = 0.4 * EIG / (1 + BetaD)
            P_Slender = (3.1415) ^ 2 * EI / ((0.65 * Val(FL_Height.Text) * 12) ^ 2)
            If Val(Factor_MomentX.Text) > 0 Then
                        Cm = 0.6 + 0.4 * (Factor_MomentX.Text / Factor_MomentX.Text)
            Else
                        Cm = 0.6
            End If
            Delta = Cm / (1 - (Factor_Axial.Text / (0.75 * P_Slender / 1000)))
            If Delta < 1 Then
                        Delta = 1
            End If
End If

' Output for the slenderness check or not
If Val(KLR.Text) <= Val(MAX_KLR.Text) Then
            Check_slender.Caption = "Slenderness check is not Required."
            Check_slender.ForeColor = &HFF0000
Else
            Check_slender.Caption = "Slenderness check is Required."
            Check_slender.ForeColor = &HFF
End If
'e/h line movement
If Val(Designed_eh.Text) > 0.1 Then
            eh_V = (2 / Val(Designed_eh.Text))
Else
eh_V = 20
End If
eh_line.X2 = 2
```

```
eh_line.Y2 = -eh_V

' <<<<<<<<<<<<<<<<<<<<<<<<   Calculate Pu and Mu   >>>>>>>>>>>>>>>>>>>>>>>>>>>
fy = Val(Steel_type.Text) / 1000
fcp = Val(Conc_type.Text) / 1000
b = Val(Column_Height.Text)
h = Val(Column_Width.Text)
dp = 2.5
tas = Val(STArea.Text) / 2
cas = Val(STArea.Text) / 2
eccentricity = Val(E_Value.Text)
z1 = 0.7225 * fcp * b
z2 = eccentricity + ((h - 2 * dp) / 2)
z3 = h - (2 * dp)
c2 = cas * (fy - 0.85 * fcp)
cubica = 0.425 * z1 / z2
cubicb = (z1 - (z1 * (h - dp) / z2))
cubicc = (c2 - (c2 * (z3 / z2)) + 87 * tas)
cubicd = (-1 * (87 * tas * (h - dp)))
Cvalue = cubic_cal(cubica, cubicb, cubicc, cubicd)
sc = Cvalue
pn = z1 * sc + c2 - 87 * ((h - dp - sc) / sc) * tas
Designed_Pu.Text = Fx(0.7 * pn)
mn = Fx(pn * eccentricity / 12)
Designed_Mu.Text = Fx(0.7 * mn)
XLine.X2 = (0.7 * mn * 12 / (b * h * h))
XLine.Y1 = -(0.7 * pn / (b * h))
XLine.Y2 = -(0.7 * pn / (b * h))
Yline.X1 = (0.7 * mn * 12 / (b * h * h))
Yline.X2 = (0.7 * mn * 12 / (b * h * h))
Yline.Y2 = -(0.7 * pn / (b * h))

'<<<<<<<<<<<<<<<<<<<<<<<< P_M Interaction Diagram drawing >>>>>>>>>>>>>>>>>>>>
PM_DRAW.Cls
Graph_Ratio = 0.02
Do
        eccentricity = Val(Column_Width.Text) / 10
        Do
                fy = Val(Steel_type.Text) / 1000
                fcp = Val(Conc_type.Text) / 1000
                b = Val(Column_Height.Text)
                h = Val(Column_Width.Text)
                dp = Val(Column_Width.Text) / 10
                tas = 0.5 * Graph_Ratio * (Column_Height.Text * Column_Width.Text)
                cas = 0.5 * Graph_Ratio * (Column_Height.Text * Column_Width.Text)
                z1 = 0.7225 * fcp * b
                z2 = eccentricity + ((h - 2 * dp) / 2)
                z3 = h - (2 * dp)
                c2 = cas * (fy - 0.85 * fcp)
                cubica = 0.425 * z1 / z2
                cubicb = (z1 - (z1 * (h - dp) / z2))
                cubicc = (c2 - (c2 * (z3 / z2)) + 87 * tas)
                cubicd = (-1 * (87 * tas * (h - dp)))
                Cvalue = cubic_cal(cubica, cubicb, cubicc, cubicd)
                sc = Cvalue
```

```
            pn = z1 * sc + c2 - 87 * ((h - dp - sc) / sc) * tas
            mn = Fx(pn * eccentricity / 12)
            PMY = (0.7 * pn / (b * h))
            PMX = (0.7 * mn * 12 / (b * h * h))
            PM_DRAW.Circle (PMX, -PMY), 0.003, &HFF0000
            eccentricity = eccentricity + 0.1
        Loop Until eccentricity > Val(Column_Width.Text)
        Graph_Ratio = Graph_Ratio + 0.01
Loop Until Graph_Ratio > 0.05
Column_Tab.Tab = 3
End Sub

Private Sub Step_04_Click()
Column_Tab.Tab = 4
End Sub

Private Sub Step_05_Click()
Column_Tab.Tab = 5
End Sub

Private Sub STRatio_Change()
Designed_STR.Text = STRatio.Text
End Sub

Private Sub Tie_Spacing_Change()
Barspacing_Info.Text = Tie_Spacing.Text
End Sub

Private Sub Tri_Lx_GotFocus()
selectfield Tri_Lx
End Sub

Private Sub Tri_Ly_gotfocus()
selectfield Tri_Ly
End Sub

' <<<<<<<<<<<<<< Cubic Equation >>>>>>>>>>>>>>>>>
Function cubic_cal(a, b, c, d) As Double
fz = ((3 * c / a) - (b * b / (a * a))) / 3
gz = ((2 * b ^ 3 / (a ^ 3)) - (9 * b * c / (a ^ 2)) + (27 * d / a)) / 27
hz = (gz ^ 2 / 4) + (fz ^ 3 / 27)
rz = -1 * (gz / 2) + Sqr(hz)
sz = rz ^ (1 / 3)
tz = -(gz / 2) - (Sqr(hz))
If tz > 0 Then
        uz = (tz) ^ (1 / 3)
Else
        tabs = Abs(tz)
        unew = (tabs) ^ (1 / 3)
        uz = -unew
End If
x1z = (sz + uz) - (b / (3 * a))
cubic_cal = x1z
End Function
Function Fx(a) As Double
```

```
nwval = a '* 1000
Fx = Fix(nwval) '/ 1000
End Function

Private Sub NullD(ByVal Zero_Divide As TextBox)
If Zero_Divide.Text = "" Then
            Zero_Divide.Text = 0
            Zero_Divide.SelStart = 0
            Zero_Divide.SelLength = 5
End If
End Sub
```

## *3. One-way Solid Slab Module*

```
' <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
'  Reinforced Concrete Structure Designer (RCSD)
' <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
'  ONE-WAY SOLID SLAB MODULE
' <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
'  University of Southern California, School of Architecture, Master of Building Science
'  Copyright 2002 by the University of Southern California and Kang-Kyu Choi
'  All right reserved
'  Contact Author: Kang-Kyu Choi  < hydrofall@hotmail.com >
'  RCSD is a reinforced concrete structure design assistant tool for learner.
' <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
Option Explicit
Dim Min_Thickness, EX_Span, IN_Span, E_Thickness, Cal_Area, Spacing, Mainbar_Size As Double
Dim MA_a, MA_C, MA_R, MB_a, MB_C, MB_R, MC_a, MC_C, MC_R, ME_a, ME_C, ME_R As
Double
Dim Max_Spacing, Max_Steel, V_Ultimate, , loadselectval, V_CE, Max_Ratio, BetaONE,
Min_DTK, Service_Load, Service_Moment As Double
Dim N_Value, N_Axis, I_CR, I_G, I_E, M_CR, CaseBar, Cal_Space, ShortD_DL, ShortD_LL,
Long_D, Long_Multi, Long_R, Long_Ratio As Double

Private Sub Barnum_MA_Click()
CaseBar = Barnum_MA.ListIndex

Select Case CaseBar
        Case 0:
                Mainbar_Size = 0.11
        Case 1:
                Mainbar_Size = 0.2
        Case 2:
                Mainbar_Size = 0.31
        Case 3:
                Mainbar_Size = 0.44
        Case 4:
                Mainbar_Size = 0.6
        Case 5:
                Mainbar_Size = 0.79
        Case 6:
                Mainbar_Size = 1
        Case 7:
                Mainbar_Size = 1.27
        Case 8:
                Mainbar_Size = 1.56
        Case 9:
                Mainbar_Size = 2.25
        Case 10:
                Mainbar_Size = 4
End Select

Cal_Space = 1

Do
        Cal_Area = Mainbar_Size * 12 / Cal_Space
```

```
        Cal_Space = Cal_Space + 1
Loop Until Cal_Area < Val(AS_MA.Text)

Space_MA.Text = Cal_Space - 2

'Warnning when the spacing is over maximum possible spacing
If Val(Space_MA.Text) > Max_Spacing Then
        MsgBox ("Reduce rebar size!!!   Rebar spacing is bigger than Maximum possible spacing.")
End If
MA_Line1.Visible = True
MA_Line2.Visible = True
End Sub

Private Sub Barnum_MB_Click()
        CaseBar = Barnum_MB.ListIndex

Select Case CaseBar
        Case 0:
                Mainbar_Size = 0.11
        Case 1:
                Mainbar_Size = 0.2
        Case 2:
                Mainbar_Size = 0.31
        Case 3:
                Mainbar_Size = 0.44
        Case 4:
                Mainbar_Size = 0.6
        Case 5:
                Mainbar_Size = 0.79
        Case 6:
                Mainbar_Size = 1
        Case 7:
                Mainbar_Size = 1.27
        Case 8:
                Mainbar_Size = 1.56
        Case 9:
                Mainbar_Size = 2.25
        Case 10:
Mainbar_Size = 4
End Select

Cal_Space = 1

Do
        Cal_Area = Mainbar_Size * 12 / Cal_Space
        Cal_Space = Cal_Space + 1
Loop Until Cal_Area < Val(AS_MB.Text)
Space_MB.Text = Cal_Space - 2

'Warnning when the spacing is over maximum possible spacing
If Val(Space_MB.Text) > Max_Spacing Then
        MsgBox ("Reduce rebar size!!!   Rebar spacing is bigger than Maximum possible spacing.")
End If
'Steel area at Maximum moment point
Max_Steel = Mainbar_Size * 12 / (Cal_Space)
```

```
MB_Line.Visible = True
End Sub

Private Sub Barnum_MC_Click()
CaseBar = Barnum_MC.ListIndex
Select Case CaseBar
        Case 0:
                Mainbar_Size = 0.11
        Case 1:
                Mainbar_Size = 0.2
        Case 2:
                Mainbar_Size = 0.31
        Case 3:
                Mainbar_Size = 0.44
        Case 4:
                Mainbar_Size = 0.6
        Case 5:
                Mainbar_Size = 0.79
        Case 6:
                Mainbar_Size = 1
        Case 7:
                Mainbar_Size = 1.27
        Case 8:
                Mainbar_Size = 1.56
        Case 9:
                Mainbar_Size = 2.25
        Case 10:
                Mainbar_Size = 4
End Select
Cal_Space = 1

Do
        Cal_Area = Mainbar_Size * 12 / Cal_Space
        Cal_Space = Cal_Space + 1
Loop Until Cal_Area < Val(AS_MC.Text)

Space_MC.Text = Cal_Space - 2

'Warnning when the spacing is over maximum possible spacing
If Val(Space_MC.Text) > Max_Spacing Then
        MsgBox ("Reduce rebar size!!!   Rebar spacing is bigger than Maximum possible spacing.")
End If
MC_Line.Visible = True
End Sub

Private Sub Barnum_ME_Click()
CaseBar = Barnum_ME.ListIndex
Select Case CaseBar
        Case 0:
                Mainbar_Size = 0.11
        Case 1:
                Mainbar_Size = 0.2
        Case 2:
                Mainbar_Size = 0.31
        Case 3:
```

```
            Mainbar_Size = 0.44
        Case 4:
            Mainbar_Size = 0.6
        Case 5:
            Mainbar_Size = 0.79
        Case 6:
            Mainbar_Size = 1
        Case 7:
            Mainbar_Size = 1.27
        Case 8:
            Mainbar_Size = 1.56
        Case 9:
            Mainbar_Size = 2.25
        Case 10:
            Mainbar_Size = 4
End Select
Cal_Space = 1

Do
        Cal_Area = Mainbar_Size * 12 / Cal_Space
        Cal_Space = Cal_Space + 1
Loop Until Cal_Area < Val(AS_ME.Text)

Space_ME.Text = Cal_Space - 2

'Warnning when the spacing is over maximum possible spacing
If Val(Space_ME.Text) > Max_Spacing Then
        MsgBox ("Reduce rebar size!!!   Rebar spacing is bigger than Maximum possible spacing.")
End If
ME_Line.Visible = True
End Sub

Private Sub Barnum_TEMP_Click()
CaseBar = Barnum_TEMP.ListIndex
Select Case CaseBar
        Case 0:
            Mainbar_Size = 0.11
        Case 1:
            Mainbar_Size = 0.2
        Case 2:
            Mainbar_Size = 0.31
        Case 3:
            Mainbar_Size = 0.44
        Case 4:
            Mainbar_Size = 0.6
        Case 5:
            Mainbar_Size = 0.79
        Case 6:
            Mainbar_Size = 1
        Case 7:
            Mainbar_Size = 1.27
        Case 8:
            Mainbar_Size = 1.56
        Case 9:
            Mainbar_Size = 2.25
```

```
        Case 10:
                Mainbar_Size = 4
End Select
Cal_Space = 1
Do
        Cal_Area = Mainbar_Size * 12 / Cal_Space
        Cal_Space = Cal_Space + 1
Loop Until Cal_Area < Val(AS_TEMP.Text)
Space_Temp.Text = Cal_Space - 2
'Warnning when the spacing is over maximum possible spacing
If Val(Space_Temp.Text) > Max_Spacing Then
        MsgBox ("Reduce rebar size!!!   Rebar spacing is bigger than Maximum possible spacing.")
End If
End Sub

Private Sub Beam1_Change()
Beam2.Text = Beam1.Text
Beam3.Text = Beam1.Text
End Sub

Private Sub Beam1_GotFocus()
selectfield Beam1
End Sub

Private Sub Beam2_GotFocus()
selectfield Beam2
End Sub

Private Sub Beam3_GotFocus()
selectfield Beam3
End Sub

Private Sub Cutted_Icon_Click()
SB_Size.Visible = False
SB_Bar.Visible = True
SB_Front.Visible = False
Line5.Visible = True
Line6.Visible = True
Line7.Visible = True
Line8.Visible = True
End Sub

Private Sub Distance1_Change()
Distance2.Text = Distance1.Text
End Sub

Private Sub Distance1_GotFocus()
selectfield Distance1
End Sub

Private Sub Distance2_GotFocus()
selectfield Distance2
End Sub
Private Sub selectfield(ByVal text_box As TextBox)
text_box.SelStart = 0
```

```
text_box.SelLength = Len(text_box.Text)
End Sub

Private Sub Dload_assist_Click()
SuperimposedDL.Show
End Sub

Private Sub Lload_assist_Click()
load_select.Visible = True
occuassist.Visible = True
End Sub

Private Sub load_select_Click()
loadselectval = load_select.ListIndex

Select Case loadselectval
        Case 0: Slab_ll.Text = 150
        Case 1: Slab_ll.Text = 100
        Case 2: Slab_ll.Text = 100
        Case 3: Slab_ll.Text = 50
        Case 4: Slab_ll.Text = 40
        Case 5: Slab_ll.Text = 150
        Case 6: Slab_ll.Text = 150
        Case 7: Slab_ll.Text = 50
        Case 8: Slab_ll.Text = 40
        Case 9: Slab_ll.Text = 40
        Case 10: Slab_ll.Text = 250
        Case 11: Slab_ll.Text = 100
        Case 12: Slab_ll.Text = 125
        Case 13: Slab_ll.Text = 125
End Select

End Sub

Private Sub LONG_ALD_Change()
LONG_TLD.Text = Val(SHORT_TLD.Text) + Val(LONG_ALD.Text)
End Sub

Private Sub LongD_Modify_Click()
'<<<<<<<<<<<<<<<< compression steel design for Long-term deflection >>>>>>>>>>>>>>>>
Long_Ratio = LongD_Modify.Text

' The Longterm deflection multiplier (DL+ sustained LL)
Long_Multi = 2 / (1 + 50 * Long_Ratio)

' The longterm deflection
Long_D = (((Val(Slab_ll.Text) / 2) + (Val(Slab_dl.Text) + (150 * Min_Thickness / 12))) /
(Val(Slab_dl.Text) + (150 * Min_Thickness / 12))) * ShortD_DL * Long_Multi
LONG_ALD.Text = Fix(Long_D * 100) / 100

If Val(Max_Deflection.Text) >= Val(SHORT_TLD.Text) Then
        If Val(Max_Deflection.Text) >= Val(LONG_TLD.Text) Then

                Deflection_Label.ForeColor = &HFF0000
```

```
                Deflection_Label.Caption = "                                        Good !!!
        Design slab thickness is adequate for the deflection! . "
                ReturnThick3.Visible = False
        Else
                Deflection_Label.ForeColor = &HFF&
                Deflection_Label.Caption = " NOT Good!!!!! Design slab thickness is NOT
        adequate for deflection. Check slab thickness!!."
                ReturnThick3.Visible = True
        End If
Else
        Deflection_Label.ForeColor = &HFF&
        Deflection_Label.Caption = " NOT Good!!!!! Design slab thickness is NOT adequate for
deflection. Check slab thickness!!."
        ReturnThick3.Visible = True
End If
If Val(MinTK_DESIGN.Text) >= Val(MinTK_MO.Text) Then
        If Val(MinTK_DESIGN.Text) >= Val(MinTK_ACI.Text) Then
                TK_LABEL.ForeColor = &HFF0000
                ThickTXT.ForeColor = &HFF0000
                ThickTXT.Caption = "                GOOD!                Design slab
        thickness is adequate for the ACI Code and Moment! . "
                ReturnThick2.Visible = False
        Else
                If    Val(Max_Deflection.Text)    >=    Val(SHORT_TLD.Text)    And
        Val(Max_Deflection.Text) >= Val(LONG_TLD.Text) Then
                        TK_LABEL.ForeColor = &HFF0000
                        ThickTXT.ForeColor = &HFF0000
                        ThickTXT.Caption = "                                GOOD!
                Design slab thickness is adequate for the ACI Code and Moment!.
                Deflection is checked!!! "
                        ReturnThick2.Visible = False
                Else
                        TK_LABEL.ForeColor = &HFF&
                        ThickTXT.ForeColor = &HFF&
                        ThickTXT.Caption = "                            NOT Good!
                Design slab thickness is not adequate! Check slab thickness and deflection!."
                        ReturnThick2.Visible = True
                End If
        End If
Else
        TK_LABEL.ForeColor = &HFF&
        ThickTXT.ForeColor = &HFF&
        ThickTXT.Caption = "                NOT Good!                Design slab
thickness is not adequate! Check slab thickness!."
        ReturnThick2.Visible = True
End If
End Sub

Private Sub Material_assist_Click()
MsgBox " Typical Steel strength = 60000 psi " + vbCrLf + " Typical Conc. Strength = 4000 or 3000
psi ", vbInformation, "Typical Material Strength"
End Sub

Private Sub Min_TK_Change()
MinTK_DESIGN.Text = Val(Min_TK.Text)
```

```
End Sub

Private Sub ReturnThick_Click()
Slab_Tab.Tab = 1
End Sub

Private Sub ReturnThick2_Click()
Slab_Tab.Tab = 1
End Sub

Private Sub ReturnThick3_Click()
Slab_Tab.Tab = 1
End Sub

Private Sub SECTION_ICON_Click()
SB_Size.Visible = False
SB_Bar.Visible = False
SB_Front.Visible = True
End Sub

Private Sub Size_Icon_Click()
SB_Size.Visible = True
SB_Bar.Visible = False
SB_Front.Visible = False
End Sub

Private Sub step_01_Click()
' Clear Span length (inch)
EX_Span = Val(Distance1.Text) * 12 - (Val(Beam1.Text) / 2) - (Val(Beam2.Text) / 2)
IN_Span = Val(Distance2.Text) * 12 - (Val(Beam2.Text) / 2) - (Val(Beam3.Text) / 2)

' Compute the minimum slab thickness
If Val(Distance1.Text) > Val(Distance2.Text) Then
        Min_Thickness = Distance1.Text * 12 / 24
Else
        Min_Thickness = Distance2.Text * 12 / 24
End If

Min_TK.Text = Min_Thickness
MinTK_ACI.Text = Min_Thickness

' factored load (kips/ft)
Factor_Load.Text = (1.4 * (Val(Slab_dl.Text) + (150 * Min_Thickness / 12)) + 1.7 * (Slab_ll.Text)) /
1000

' Compute the moments
M_A.Text = Fix(((Factor_Load.Text / 12) * (EX_Span ^ 2) / 24) * 100) / 100
M_B.Text = Fix(((Factor_Load.Text / 12) * (EX_Span ^ 2) / 14) * 100) / 100
M_C.Text = Fix(((Factor_Load.Text / 12) * (EX_Span ^ 2) / 10) * 100) / 100
M_D.Text = Fix(((Factor_Load.Text / 12) * (IN_Span ^ 2) / 11) * 100) / 100
M_E.Text = Fix(((Factor_Load.Text / 12) * (EX_Span ^ 2) / 16) * 100) / 100
Slab_Tab.Tab = 1
End Sub
Private Sub Step_02_Click()
E_Thickness = Val(Min_TK.Text) - 1
```

```
'<<<< maximum tension steel ratio >>>>
BetaONE = 0.85 - 0.05 * ((Val(Conc_type.Text) - 4000) / 1000)
If BetaONE > 0.85 Then
        BetaONE = 0.85
End If

If BetaONE < 0.65 Then
        BetaONE = 0.65
End If

Max_Ratio = (0.75 * 0.85 * BetaONE * (Conc_type.Text / Steel_type.Text) * (87000 / (87000 +
Steel_type.Text)))

' compute MA steel Area
MA_a = ((10.2 * (Conc_type.Text / 1000) * E_Thickness) - Sqr(((10.2 * (Conc_type.Text / 1000) *
E_Thickness) ^ 2) - (20.4 * (Conc_type.Text / 1000) * (M_A.Text / 0.9)))) / (10.2 * (Conc_type.Text
/ 1000))
MA_C = 0.85 * (Conc_type.Text / 1000) * MA_a * 12
MA_R = (MA_C / (Steel_type.Text / 1000)) / (12 * E_Thickness)
If MA_R > Max_Ratio Then
        AS_MA = Max_Ratio * 12 * E_Thickness
Else
        AS_MA = Fix((MA_C / (Steel_type.Text / 1000)) * 1000) / 1000
End If

' compute MB steel Area
MB_a = ((10.2 * (Conc_type.Text / 1000) * E_Thickness) - Sqr(((10.2 * (Conc_type.Text / 1000) *
E_Thickness) ^ 2) - (20.4 * (Conc_type.Text / 1000) * (M_B.Text / 0.9)))) / (10.2 * (Conc_type.Text
/ 1000))
MB_C = 0.85 * (Conc_type.Text / 1000) * MB_a * 12
MB_R = (MB_C / (Steel_type.Text / 1000)) / (12 * E_Thickness)

If MA_R > Max_Ratio Then
        AS_MB = Max_Ratio * 12 * E_Thickness
Else
        AS_MB = Fix((MB_C / (Steel_type.Text / 1000)) * 1000) / 1000
End If

' compute MC steel Area
MC_a = ((10.2 * (Conc_type.Text / 1000) * E_Thickness) - Sqr(((10.2 * (Conc_type.Text / 1000) *
E_Thickness) ^ 2) - (20.4 * (Conc_type.Text / 1000) * (M_C.Text / 0.9)))) / (10.2 * (Conc_type.Text
/ 1000))
MC_C = 0.85 * (Conc_type.Text / 1000) * MC_a * 12
MC_R = (MC_C / (Steel_type.Text / 1000)) / (12 * E_Thickness)

If MA_R > Max_Ratio Then
   AS_MC = Max_Ratio * 12 * E_Thickness
Else
AS_MC = Fix((MC_C / (Steel_type.Text / 1000)) * 1000) / 1000
End If
```

```
' compute ME steel Area
ME_a = ((10.2 * (Conc_type.Text / 1000) * E_Thickness) - Sqr(((10.2 * (Conc_type.Text / 1000) * E_Thickness) ^ 2) - (20.4 * (Conc_type.Text / 1000) * (M_E.Text / 0.9)))) / (10.2 * (Conc_type.Text / 1000))
ME_C = 0.85 * (Conc_type.Text / 1000) * ME_a * 12
ME_R = (ME_C / (Steel_type.Text / 1000))

If MA_R > Max_Ratio Then
        AS_ME = Max_Ratio * 12 * E_Thickness
Else
        AS_ME = Fix((ME_C / (Steel_type.Text / 1000)) * 1000) / 1000
End If

'Compute Temperature Steel
If Val(Steel_type.Text) > 50000 Then
        If Val(Steel_type.Text) = 60000 Then
                AS_TEMP.Text = 0.0018 * 12 * Val(Min_TK.Text)
        Else
                AS_TEMP.Text = (0.0018 * (60000) / Val(Steel_type.Text)) * 12 *
        Val(Min_TK.Text)
        End If
Else
        AS_TEMP.Text = 0.002 * 12 * Val(Min_TK.Text)
End If
'Maximun possible spacing
If 5 * (Min_TK.Text) > 18 Then
        Max_Spacing = 18
Else
        Max_Spacing = 5 * (Min_TK.Text)
End If
MX_SPacing.Text = Max_Spacing
Slab_Tab.Tab = 2
End Sub

Private Sub Step_03_Click()
' <<<<<<<<<<<<<<<<<<<<<<<<<< shear check >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
V_Ultimate = (1.15 / 2) * Val(Factor_Load.Text / 12) * EX_Span - (Val(Factor_Load.Text / 12) * E_Thickness / 12)
V_CE = 0.85 * 2 * Sqr(Conc_type.Text) * 12 * E_Thickness / 1000
Vu.Text = Fix(V_Ultimate * 100) / 100
oVc.Text = Fix(V_CE * 100) / 100

If V_Ultimate > V_CE Then
        ShearJurge.Caption = "NOT GOOD.                    Change the footing thickness"
        ShearJurge.ForeColor = &HFF&
        ShearBad.Visible = True
        ShearOK.Visible = False
        ShearTXT.Caption = "Shear Strength on the critical section is Greater than the shear strength of concrete. "
        ReturnThick.Visible = True
Else
        ShearJurge.Caption = "GOOD"
        ShearJurge.ForeColor = &HFF0000

        ShearBad.Visible = False
```

```
            ShearOK.Visible = True
            ShearTXT.Caption = " Shear Strength on the critical section is Less than the shear strength of concrete."
            ReturnThick.Visible = False
End If

'<<<<<<<<<<<<<<<< Deflection check >>>>>>>>>>>>>>>>>>>>>>>
' modular ratio n
If Val(Conc_type.Text) > 4000 Then
            If Val(Conc_type.Text) = 4000 Then
                        N_Value = 8
            Else
                        N_Value = 7
            End If
Else
            N_Value = 9
End If

' Neutral-axis location
N_Axis = N_Value * Max_Steel * ((Sqr(1 + (2 * 12 * E_Thickness / (N_Value * Max_Steel)))) - 1) / 12
' The moment inertia of the cracked section
I_CR = (12 * (N_Axis ^ 3) / 3) + ((N_Value * Max_Steel) * (E_Thickness - N_Axis) ^ 2)
'The moment inertia of the gross section
I_G = (Val(Min_TK.Text) ^ 3) * 12 / 12
' the moment would initially crack the cross section
M_CR = 7.5 * Sqr(Conc_type.Text / 1000000) * I_G / (Min_TK.Text * 12 / 2)
' The effective moment of inertia
I_E = (((M_CR / M_C) ^ 3 * I_G) + ((1 - ((M_CR / M_C) ^ 3))) * I_CR)
Service_Load = ((Val(Slab_dl.Text) + (150 * Min_Thickness / 12))) / 1000
' THE Immediate dead load deflection
ShortD_DL = (5 * (EX_Span ^ 3) * (Service_Load * 1000 * EX_Span / 12)) / ((384 * 57 * Sqr(Conc_type.Text) * 1000 * I_G))
ShortD_LL = (Val(Slab_ll.Text) / (Val(Slab_dl.Text) + (150 * Min_Thickness / 12))) * ShortD_DL
' The Longterm deflection multiplier (DL+ sustained LL)
Long_Multi = 2 / (1 + 50 * 0)
' The longterm deflection
Long_D = (((Val(Slab_ll.Text) / 2) + (Val(Slab_dl.Text) + (150 * Min_Thickness / 12))) / (Val(Slab_dl.Text) + (150 * Min_Thickness / 12))) * ShortD_DL * Long_Multi
SHORT_DLD.Text = Fix(ShortD_DL * 100) / 100
SHORT_LLD.Text = Fix(ShortD_LL * 100) / 100
SHORT_TLD.Text = Val(SHORT_DLD.Text) + Val(SHORT_LLD.Text)
LONG_ALD.Text = Fix(Long_D * 100) / 100

' maximum allowable deflection
Max_Deflection.Text = Distance1.Text * 12 / 240

If Val(Max_Deflection.Text) >= Val(SHORT_TLD.Text) Then
            If Val(Max_Deflection.Text) >= Val(LONG_TLD.Text) Then
                        Deflection_Label.ForeColor = &HFF0000
                        Deflection_Label.Caption = "                                    Good !!!
            Design slab thickness is adequate for the deflection! . "
                        ReturnThick3.Visible = False
            Else
                        Deflection_Label.ForeColor = &HFF&
```

```
		Deflection_Label.Caption = " NOT Good!!!!! Design slab thickness is NOT
	adequate for deflection. Check slab thickness!!."
		ReturnThick3.Visible = True
	End If
Else
	Deflection_Label.ForeColor = &HFF&
	Deflection_Label.Caption = " NOT Good!!!!! Design slab thickness is NOT adequate for
deflection. Check slab thickness!!."
	ReturnThick3.Visible = True
End If
'<<<<<<<<<<<<<<<<<<<<<<<<< Minimum slab thickness check>>>>>>>>>>>>>>>>>>>>>>>
Min_DTK = Val(M_C.Text) / ((0.9 * Max_Ratio * (Steel_type.Text / 1000) * (1 - 0.59 * Max_Ratio
* Steel_type.Text / Conc_type.Text)) * 12)
MinTK_MO.Text = Fix(Sqr(Min_DTK) * 100) / 100 + 1

If Val(MinTK_DESIGN.Text) >= Val(MinTK_ACI.Text) And Val(MinTK_MO.Text) Then
	TK_LABEL.ForeColor = &HFF0000
	ThickTXT.ForeColor = &HFF0000
	ThickTXT.Caption = "			GOOD!			Design slab thickness is
adequate for the ACI Code and Moment! . "
	ReturnThick2.Visible = False
Else
	If Val(Max_Deflection.Text) >= Val(SHORT_TLD.Text) Then
		If Val(Max_Deflection.Text) >= Val(LONG_TLD.Text) Then
			TK_LABEL.ForeColor = &HFF0000
			ThickTXT.ForeColor = &HFF0000
			ThickTXT.Caption = "		GOOD!			Design slab
		thickness is adequate for the ACI Code and Moment!.		Deflection is
		checked!!! "
			ReturnThick2.Visible = False
		Else
			TK_LABEL.ForeColor = &HFF&
			ThickTXT.ForeColor = &HFF&
			ThickTXT.Caption = "		NOT Good!			Design slab
		thickness is not adequate! Check slab thickness and deflection!."
			ReturnThick2.Visible = True
		End If
	Else
		TK_LABEL.ForeColor = &HFF&
		ThickTXT.ForeColor = &HFF&
		ThickTXT.Caption = "			NOT Good!			Design slab
	thickness is not adequate! Check slab thickness and deflection!."
	ReturnThick2.Visible = True
	End If
End If
Slab_Tab.Tab = 3
End Sub
Private Sub Step_04_Click()
Slab_Tab.Tab = 4
End Sub
```

## 4. Two-way Slab Module

```
'<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
' Reinforced Concrete Structure Designer (RCSD)
'<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
' TWO-WAY SLAB MODULE
'<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
' University of Southern California, School of Architecture, Master of Building Science
' Copyright 2002 by the University of Southern California and Kang-Kyu Choi
' All right reserved
' Contact Author: Kang-Kyu Choi < hydrofall@hotmail.com >
' RCSD is a reinforced concrete structure design assistant tool for learner.
'<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
Option Explicit
Dim loadselectval, Slab_Shape, Spandrel_shape, Slab_Type, Min_SlabTK As Double
Dim Factor_DL, Factor_LL, R_MO, R_DLoad, R_lLoad, E_Thickness, K_Bar, Steel_Rreq,
Steel_Rmin, Steel_Rmax, Req_Steel, Req_Csteel, BetaOne As Double
Dim N_Value, N_Axis, I_CR, I_G, I_E, M_CR, ShortD_DL, ShortD_LL, Long_D, Long_Multi,
Long_R, Long_Ratio As Double

Private Sub Command1_Click()
MsgBox "These steel ratios are calculated using Limiting steel ratios for flexural design.",
vbInformation, "Information"
End Sub

Private Sub Dload_assist_Click()
SuperimposedDL1.Show
End Sub

Private Sub Lload_assist_Click()
load_select.Visible = True
occuassist.Visible = True
End Sub

Private Sub load_select_click()
loadselectval = load_select.ListIndex
Select Case loadselectval
        Case 0: Slab_ll.Text = 150
        Case 1: Slab_ll.Text = 100
        Case 2: Slab_ll.Text = 100
        Case 3: Slab_ll.Text = 50
        Case 4: Slab_ll.Text = 40
        Case 5: Slab_ll.Text = 150
        Case 6: Slab_ll.Text = 150
        Case 7: Slab_ll.Text = 50
        Case 8: Slab_ll.Text = 40
        Case 9: Slab_ll.Text = 40
        Case 10: Slab_ll.Text = 250
        Case 11: Slab_ll.Text = 100
        Case 12: Slab_ll.Text = 125
        Case 13: Slab_ll.Text = 125
End Select
End Sub
```

```
Private Sub LONG_ALD_Change()
LONG_TLD.Text = Val(SHORT_TLD.Text) + Val(LONG_ALD.Text)
End Sub

Private Sub Material_assist_Click()
MsgBox " Typical Steel strength = 60000 psi " + vbCrLf + " Typical Conc. Strength = 4000 or 3000
psi ", vbInformation, "Typical Material Strength"
End Sub

Private Sub Moment_Cal_Assist_Click()
MsgBox "Modified load can be calculated by applying a Reduction factor to the service load." +
vbCrLf + "" + vbCrLf + "The Reduction factor can be calculated using " + vbCrLf + "        R = (LX^4)
/ ((Lx^4)+(Ly^4)) ", vbInformation, "Information"
End Sub

Private Sub RE_Deflection_Click()
E_Thickness = Val(SLAB_THICKNESS.Text) - 1.5
Req_Steel = Fix((Val(Ten_ST.Text) * 12 * E_Thickness) * 100) / 100
Req_CSteel = Fix((Val(Com_ST.Text) * 12 * E_Thickness) * 100) / 100

' modular ratio n
If Val(Conc_type.Text) > 4000 Then
        If Val(Conc_type.Text) = 4000 Then
                N_Value = 8
        Else
                N_Value = 7
        End If
Else
        N_Value = 9
End If

' Neutral-axis location
If Val(Com_ST.Text) > 0 Then
        N_Axis = N_Value * (Req_Steel + Req_CSteel) * ((Sqr(1 + (2 * 12 * (E_Thickness + 1) /
(N_Value * (Req_Steel + Req_CSteel))))) - 1) / (12)
Else
        N_Axis = N_Value * (Req_Steel) * (Sqr(1 + (2 * 12 * E_Thickness / (N_Value *
(Req_Steel))))) / 12
End If

' The moment inertia of the cracked section
I_CR = (12 * (N_Axis ^ 3) / 3) + ((N_Value * (Req_Steel)) * (E_Thickness - N_Axis) ^ 2) +
((N_Value * (Req_CSteel)) * (N_Axis - E_Thickness) ^ 2)

'The moment inertia of the gross section
I_G = (SLAB_THICKNESS.Text ^ 3) * 12 / 12

' the moment would initially crack the cross section
M_CR = 7.5 * Sqr(Conc_type.Text / 1000000) * I_G / (SLAB_THICKNESS.Text * 12 / 2)

' The effective moment of inertia
I_E = (((M_CR / Val(MO_TL.Text)) ^ 3 * I_G) + ((1 - ((M_CR / Val(MO_TL.Text)) ^ 3))) * I_CR)
```

```
' THE Immediate dead load deflection
ShortD_DL = (5 * Val(MO_DL.Text) * ((DIM_LX.Text) ^ 2) * 1728) / (48 * 57 * Sqr(Conc_type.Text) * I_E)
ShortD_LL = (Val(MO_LL.Text) / Val(MO_DL.Text)) * ShortD_DL
' The Longterm deflection multiplier (DL+ sustained LL)
Long_Multi = 2 / (1 + 50 * Val(Com_ST.Text))
' The longterm deflection
Long_D = (((Val(MO_LL.Text) / 2) + (Val(MO_DL.Text))) / (Val(MO_DL.Text))) * ShortD_DL * Long_Multi
SHORT_DLD.Text = Fix(ShortD_DL * 100) / 100
SHORT_LLD.Text = Fix(ShortD_LL * 100) / 100
SHORT_TLD.Text = Val(SHORT_DLD.Text) + Val(SHORT_LLD.Text)
LONG_ALD.Text = Fix(Long_D * 100) / 100

' maximum allowable deflection
Max_Deflection.Text = Fix((DIM_LX.Text * 12 / 240) * 100) / 100
If Val(Max_Deflection.Text) > Val(SHORT_TLD.Text) And Val(Max_Deflection.Text) > Val(LONG_TLD.Text) Then
        Jurge_D.Caption = "    GOOD ! : )     "
        Jurge_D.ForeColor = &HFF0000
Else
        Jurge_D.Caption = "   NOT GOOD ! : (     "
        Jurge_D.ForeColor = &HFF&
End If
End Sub

Private Sub REC_OP_Click()
If REC_OP.Value = True Then
        SQR_DIM.Visible = False
        REC_DIM.Visible = True
End If
End Sub

Private Sub Rectangular_Type_Click()
If Rectangular_Type.Value = True Then
        Slab_Shape = 2
End If
End Sub

Private Sub Span_Assist_Click()
MsgBox "The Longer of the two clear spans is always considered for calculation.", vbInformation, "Information"
End Sub

Private Sub SpanNo_Click()
If SpanNo.Value = True Then
        YesSpandrel.Visible = False
        NOSpandrel.Visible = True
        Spandrel_shape = 2
End If
End Sub


Private Sub SpanYes_Click()
If SpanYes.Value = True Then
```

```
        YesSpandrel.Visible = True
        NOSpandrel.Visible = False
        Spandrel_shape = 1
End If
End Sub

Private Sub SQR_OP_Click()
If SQR_OP.Value = True Then
        SQR_DIM.Visible = True
        REC_DIM.Visible = False
End If
End Sub

Private Sub Square_type_Click()
If Square_type.Value = True Then
        Slab_Shape = 1
End If
End Sub

Private Sub step_01_Click()
If Type01.Value = True Then
        Slab_Type = 1
        Shape_Frame.Visible = True
        Spandrel_Frame.Visible = False
        Slab_Shape = 1
End If

If Type02.Value = True Then
        Slab_Type = 2
        Shape_Frame.Visible = False
        Spandrel_Frame.Visible = True
        Spandrel_shape = 1
End If

If Type03.Value = True Then
        Slab_Type = 3
        Shape_Frame.Visible = False
        Spandrel_Frame.Visible = True
        Spandrel_shape = 1
End If

MinSlab_Tab.Tab = 1
End Sub

Private Sub step_02_Click()
'two-way slab (square panel)
If Slab_Type = 1 And Slab_Shape = 1 Then
        Min_SlabTK = 8.5 * Val(Clear_Span.Text) / 28.5
        Tway_MinTK.Text = Format(Min_SlabTK, "##.#")
End If


'two-way slab (rectangular panel)
If Slab_Type = 1 And Slab_Shape = 2 Then
```

```
		Min_SlabTK = 6.9 * Val(Clear_Span.Text) / 26.5
		Tway_MinTK.Text = Format(Min_SlabTK, "##.#")
End If

' flat plate with spandrel beams
If Slab_Type = 2 And Spandrel_shape = 1 Then
		Min_SlabTK = 10 * Val(Clear_Span.Text) / 27.5
		If Min_SlabTK >= 4 Then
				Tway_MinTK.Text = Format(Min_SlabTK, "##.#")
		Else
				Tway_MinTK.Text = 4
		End If
End If

' flat plate without spandrel beams
If Slab_Type = 2 And Spandrel_shape = 2 Then
		Min_SlabTK = 9 * Val(Clear_Span.Text) / 23 + 0.1
		If Min_SlabTK >= 5 Then
				Tway_MinTK.Text = Format(Min_SlabTK, "##.#")
		Else
				Tway_MinTK.Text = 5
		End If
End If

' flat slab with spandrel beams
If Slab_Type = 3 And Spandrel_shape = 1 Then
		Min_SlabTK = 9.25 * Val(Clear_Span.Text) / 28
		If Min_SlabTK >= 4 Then
				Tway_MinTK.Text = Format(Min_SlabTK, "##.#")
		Else
				Tway_MinTK.Text = 4
		End If
End If

' flat slab without spandrel beams
If Slab_Type = 3 And Spandrel_shape = 2 Then
		Min_SlabTK = 10 * Val(Clear_Span.Text) / 27.5
		If Min_SlabTK >= 4 Then
				Tway_MinTK.Text = Format(Min_SlabTK, "##.#")
		Else
				Tway_MinTK.Text = 4
		End If
End If

Mark.Left = Val(Clear_Span.Text) - 0.9
Mark.Top = (14 - Val(Tway_MinTK.Text)) - 0.2
MinSlab_Tab.Tab = 2
End Sub

Private Sub Step_03_Click()
SLAB_THICKNESS.Text = Val(Tway_MinTK.Text)


If Slab_Shape = 2 Then
		DIM_LX.Text = Val(Clear_Span.Text)
```

```
            DIM_LY.Text = Val(Clear_Span.Text) / 2
            REC_OP.Value = True
Else
            DIM_LX.Text = Val(Clear_Span.Text)
            DIM_LY.Text = Val(Clear_Span.Text)
            SQR_OP.Value = True
End If

' Calcucate the Service load including self-weight
Factor_DL = ((Val(Slab_dl.Text) + (150 * Tway_MinTK / 12)) / 1000)
Factor_LL = ((Slab_ll.Text)) / 1000

' Calculate modified service load by simplified method ( Prof. G.G. Schierle )
If Slab_Shape = 2 Then
            R_MO = ((Val(DIM_LX.Text)) ^ 4) / (((Val(DIM_LX.Text)) ^ 4) + ((Val(DIM_LY.Text)) ^
4))
            R_DLoad = R_MO * Factor_DL
            R_lLoad = R_MO * Factor_LL
Else
            R_DLoad = Factor_DL / 2
            R_lLoad = Factor_LL / 2
End If

MO_DEADLOAD.Text = Fix(R_DLoad * 1000) / 1000
MO_LIVELOAD.Text = Fix(R_lLoad * 1000) / 1000
MO_TOTALLOAD.Text = Fix((R_lLoad + R_DLoad) * 1000) / 1000

' Calculate the modified moment
MO_DL.Text = Fix((R_DLoad * (Val(DIM_LX.Text) ^ 2) / 8) * 100) / 100
MO_LL.Text = Fix((R_lLoad * (Val(DIM_LX.Text) ^ 2) / 8) * 100) / 100
MO_TL.Text = Val(MO_DL.Text) + Val(MO_LL.Text)

'<<<<<<<<<<<<<<< Deflection check >>>>>>>>>>>>>>>>>>>>>>
'-------------- require steel area calculate -----------------------------------
E_Thickness = Val(SLAB_THICKNESS.Text) - 1.5

'Compute Coefficient of Resistance
K_Bar = (Val(MO_TL.Text) * 12) / (0.9 * 12 * (E_Thickness ^ 2))

'Compute the required steel ratio
Steel_Rreq = (Val(Steel_type.Text) - Sqr((Steel_type.Text ^ 2) - (2.352 * (Steel_type.Text ^ 2) *
(K_Bar * 1000) / Conc_type.Text))) / (1.176 * (Steel_type.Text ^ 2) / Val(Conc_type.Text))

'Compute the minimum steel ratio
If 3 * Sqr(Conc_type.Text) >= 200 Then
            Steel_Rmin = (3 * Sqr(Conc_type.Text) / Steel_type.Text)
Else
            Steel_Rmin = (200 / Steel_type.Text)
End If

'Compute the maximum steel ratio
BetaOne = 0.85 - 0.05 * ((Val(Conc_type.Text) - 4000) / 1000)

If BetaOne > 0.85 Then
            BetaOne = 0.85
```

```
End If

If BetaOne < 0.65 Then
        BetaOne = 0.65
End If

Steel_Rmax = Fix((0.75 * 0.85 * BetaOne * (Conc_type.Text / Steel_type.Text) * (87000 / (87000 +
Steel_type.Text))) * 100) / 100

'Compute design steel area ( maximum steel / 2 )
Req_Steel = Fix(((Steel_Rmax / 2) * 12 * E_Thickness) * 100) / 100
Ten_ST.Text = Fix((Steel_Rmax / 2) * 1000) / 1000
Com_ST.Text = 0
STR_MAX.Text = Fix(Steel_Rmax * 1000) / 1000
STR_MIN.Text = Fix(Steel_Rmin * 1000) / 1000

' modular ratio n
If Val(Conc_type.Text) > 4000 Then
        If Val(Conc_type.Text) = 4000 Then
                N_Value = 8
        Else
                N_Value = 7
        End If
Else
        N_Value = 9
End If

' Neutral-axis location
N_Axis = N_Value * (Req_Steel) * (Sqr(1 + (2 * 12 * E_Thickness / (N_Value * (Req_Steel))))) / 12

' The moment inertia of the cracked section
I_CR = (12 * (N_Axis ^ 3) / 3) + ((N_Value * (Req_Steel)) * (E_Thickness - N_Axis) ^ 2)

'The moment inertia of the gross section
I_G = (SLAB_THICKNESS.Text ^ 3) * 12 / 12

' the moment would initially crack the cross section
M_CR = 7.5 * Sqr(Conc_type.Text / 1000000) * I_G / (SLAB_THICKNESS.Text * 12 / 2)

' The effective moment of inertia
I_E = (((M_CR / Val(MO_TL.Text)) ^ 3 * I_G) + ((1 - ((M_CR / Val(MO_TL.Text)) ^ 3))) * I_CR)

' THE Immediate dead load deflection
ShortD_DL = (5 * Val(MO_DL.Text) * ((DIM_LX.Text) ^ 2) * 1728) / (48 * 57 *
Sqr(Conc_type.Text) * I_E)
ShortD_LL = (Val(MO_LL.Text) / Val(MO_DL.Text)) * ShortD_DL

' The Longterm deflection multiplier (DL+ sustained LL)
Long_Multi = 2 / (1 + 50 * 0)

' The longterm deflection
Long_D = (((Val(MO_LL.Text) / 2) + (Val(MO_DL.Text))) / (Val(MO_DL.Text))) * ShortD_DL *
Long_Multi
SHORT_DLD.Text = Fix(ShortD_DL * 100) / 100
SHORT_LLD.Text = Fix(ShortD_LL * 100) / 100
```

```
SHORT_TLD.Text = Val(SHORT_DLD.Text) + Val(SHORT_LLD.Text)
LONG_ALD.Text = Fix(Long_D * 100) / 100

' maximum allowable deflection
Max_Deflection.Text = Fix((DIM_LX.Text * 12 / 240) * 100) / 100

If Val(Max_Deflection.Text) > Val(SHORT_TLD.Text) And Val(Max_Deflection.Text) > Val(LONG_TLD.Text) Then
        Jurge_D.Caption = "    GOOD ! : )     "
        Jurge_D.ForeColor = &HFF0000
Else
        Jurge_D.Caption = "   NOT GOOD ! : (    "
        Jurge_D.ForeColor = &HFF&
End If

MinSlab_Tab.Tab = 3
End Sub

Private Sub Type01_Click()
If Type01.Value = True Then
        Slab_Type = 1
End If
End Sub

Private Sub Type02_Click()
If Type02.Value = True Then
        Slab_Type = 2
End If
End Sub

Private Sub Type03_Click()
If Type03.Value = True Then
        Slab_Type = 3
End If


End Sub
```

## *5. Individual Column Footing Module*

```
'<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
'  Reinforced Concrete Structure Designer (RCSD)
'<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
'  INDIVIDUAL COLUMN FOOTING MODULE
'<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
'  University of Southern California, School of Architecture, Master of Building Science
'  Copyright 2002 by the University of Southern California and Kang-Kyu Choi
'  All right reserved
'  Contact Author: Kang-Kyu Choi  < hydrofall@hotmail.com >
'  RCSD is a reinforced concrete structure design assistant tool for learner.
'<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
Option Explicit
Dim Ft_Sw, Soil_Wt, Esoil_Pres, Rsoil_Pres, Spacing, Casebar, CaseBar1 As Double
Dim Factored_Shear, Conc_Shear, Conc_Shear1, Conc_Shear2, Conc_Shear3, Onefactored_Shear, Oneconc_Shear Rfoot_Size, Rfoot_Width, P_Ultimate, Foot_Tk,  As Double
Dim M_Design, X1, Y1, X2, Y2, St_Count, K_ER As Double
Dim E_Depth, V_Ultimate, V_C, DSoil_Pres, M_Ultimate, K_Bar, Steel_R, Steel_Rmin, Mainbar_Size, SteelDia, Tempbar_Size As Double
Dim Quantity_Bar, Alpha, Beta, Delta, Gamma, C_Factor, C_db, L_Develope As Double

Private Sub barnum_Click()
Casebar = Barnum.ListIndex
Select Case Casebar
        Case 0:
        Mainbar_Size = 0.11
        SteelDia = 0.375
        Case 1:
        Mainbar_Size = 0.2
        SteelDia = 0.5
        Case 2:
        Mainbar_Size = 0.31
        SteelDia = 0.625
        Case 3:
        Mainbar_Size = 0.44
        SteelDia = 0.75
        Case 4:
        Mainbar_Size = 0.6
        SteelDia = 0.875
        Case 5:
        Mainbar_Size = 0.79
        SteelDia = 1
        Case 6:
        Mainbar_Size = 1
        SteelDia = 1.128
        Case 7:
        Mainbar_Size = 1.27
        SteelDia = 1.27
        Case 8:
        Mainbar_Size = 1.56
        SteelDia = 1.41
        Case 9:
        Mainbar_Size = 2.25
```

```
        SteelDia = 1.693
        Case 10:
        Mainbar_Size = 4
        SteelDia = 2.257
End Select

' The number of reinforcement - Round( +0.49) Syntax for next higher integer
Main_Quantity.Text = Round((Val(Req_steel1.Text) / Mainbar_Size) + 0.49)

'Drawing the reinforcing steel with scale
Spacing = 2 * (Val(WD_TXT.Text) * 12 - 12) / (Val(Main_Quantity.Text) - 1)
Scaledraw2.Cls
Scaledraw3.Cls
St_Count = 0
Do
        X1 = -1 * (Val(WD_TXT.Text * 12) - (12 + (St_Count * Spacing)) - (SteelDia / 2)) / 2
        Y1 = (Val(WD_TXT.Text * 12) - 6) / 2
        X2 = -1 * (Val(WD_TXT.Text * 12) - (12 + (St_Count * Spacing)) + (SteelDia / 2)) / 2
        Y2 = -1 * (Val(WD_TXT.Text * 12) - 6) / 2
        Scaledraw2.Line (X1, Y1)-(X2, Y2), &HC000C0, BF
        Scaledraw2.Line (Y1, X1)-(Y2, X2), &HC000C0, BF
        Scaledraw3.Line (X1, Y1)-(X2, Y2), &HC000C0, BF
        Scaledraw3.Line (Y1, X1)-(Y2, X2), &HC000C0, BF
        St_Count = St_Count + 1
Loop Until St_Count > (Val(Main_Quantity.Text) - 1)

DR_Column2.Visible = False
DR_Column2.Visible = True
DR_Column3.Visible = False
DR_Column3.Visible = True
Result_Num.Text = Barnum.Text
Result_Num1.Text = Barnum.Text
End Sub

Private Sub Column_Thick_Change()
Result_Column.Text = Column_Thick.Text
End Sub

Private Sub Dowel_ICon_Click()
PIC_ALL.Visible = False
PIC_Clear.Visible = False
PIC_Detail.Visible = True
End Sub

Private Sub Foot_Thick_Change()
' for prevent type mismatch error by null inputing
If Foot_Thick.Text = "" Then
        Foot_Thick.Text = 0
        Foot_Thick.SelStart = 0
        Foot_Thick.SelLength = 5
End If


If Val(Foot_Thick.Text) > Val(UnderDepth_Foot.Text) Then
        Foot_Thick.Text = Val(UnderDepth_Foot.Text)
```

```
            Foot_Thick.SelStart = 0
            Foot_Thick.SelLength = 5
End If

 Foot_Tk = Val(Foot_Thick.Text)

' Drawing procedure when user change the thickness of footing
Ft_Sw = 0.15 * Foot_Tk
Soil_Wt = Wt_Earth.Text * (Val(UnderDepth_Foot.Text) - Foot_Tk) / 1000
Esoil_Pres = (Soil_Pres.Text / 1000) - Ft_Sw - Soil_Wt
Rfoot_Size = (Val(Foot_DL.Text) + Val(Foot_LL.Text)) / Esoil_Pres
Rfoot_Width = Sqr(Rfoot_Size)

If Rfoot_Width > Fix(Rfoot_Width) + 0.5 Then
            Rfoot_Width = Round(Rfoot_Width)
Else
            Rfoot_Width = Fix(Rfoot_Width) + 0.5
End If

Rfoot_Size = (Rfoot_Width) ^ 2
P_Ultimate = Val(Factor_Load.Text) / Rfoot_Size
E_Depth = (Foot_Tk * 12) - 4
Factored_Shear = P_Ultimate * (Rfoot_Size - (((Val(Column_Thick.Text) + E_Depth) / 12) ^ 2))
Conc_Shear1 = (2 + 4) * (Sqr(Val(Conc_type.Text))) * (Val(Column_Thick.Text) + E_Depth) * 4 *
E_Depth
Conc_Shear2 = ((40 * E_Depth) / ((Val(Column_Thick.Text) + E_Depth) * 4) + 2) *
(Sqr(Val(Conc_type.Text))) * (Val(Column_Thick.Text) + E_Depth) * 4 * E_Depth
Conc_Shear3 = (4) * (Sqr(Val(Conc_type.Text))) * (Val(Column_Thick.Text) + E_Depth) * 4 *
E_Depth

If Conc_Shear2 > Conc_Shear3 Then
            Conc_Shear = 0.85 * Conc_Shear3 / 1000
Else
            Conc_Shear = 0.85 * Conc_Shear2 / 1000
End If

Onefactored_Shear = P_Ultimate * Rfoot_Width * (((Rfoot_Width - (Val(Column_Thick.Text) / 12) -
(2 * E_Depth / 12))) / 2)
Oneconc_Shear = 0.85 * ((2) * (Sqr(Val(Conc_type.Text))) * Rfoot_Width * 12 * E_Depth) / 1000
If Conc_Shear > Factored_Shear Then
            If Oneconc_Shear > Onefactored_Shear Then
                        WD_TXT.Text = Format(Rfoot_Width, "####.##")
            End If
End If
ScaleDraw.ScaleWidth = 2 * Rfoot_Width
ScaleDraw.ScaleHeight = 2 * Rfoot_Width * 7 / 9
ScaleDraw.ScaleTop = -(14 * Rfoot_Width / 18)
ScaleDraw.ScaleLeft = -1 * Rfoot_Width
DR_Footing.Height = Rfoot_Width
DR_Footing.Width = Rfoot_Width
DR_Column.Height = Column_Thick.Text / 12
DR_Column.Width = Column_Thick.Text / 12
DR_Footing.Top = -1 * Rfoot_Width / 2
DR_Footing.Left = -1 * Rfoot_Width / 2
DR_Column.Top = -1 * Column_Thick.Text / 24
```

```
DR_Column.Left = -1 * Column_Thick.Text / 24

'Drawing the plan view of footing and column in Tab3
Scaledraw2.ScaleWidth = 1.2 * Rfoot_Width * 12
Scaledraw2.ScaleHeight = 1.2 * Rfoot_Width * 12
Scaledraw2.ScaleTop = -0.6 * Rfoot_Width * 12
Scaledraw2.ScaleLeft = -0.6 * Rfoot_Width * 12
DR_Footing2.Height = Rfoot_Width * 12
DR_Footing2.Width = Rfoot_Width * 12
DR_Column2.Height = Column_Thick.Text
DR_Column2.Width = Column_Thick.Text
DR_Footing2.Top = -1 * Rfoot_Width * 12 / 2
DR_Footing2.Left = -1 * Rfoot_Width * 12 / 2
DR_Column2.Top = -1 * Column_Thick.Text / 2
DR_Column2.Left = -1 * Column_Thick.Text / 2
Result_TK.Text = Foot_Thick.Text
End Sub

Private Sub Foot_Thick_LostFocus()
' warnning massage for zero thickness footing
If Foot_Thick.Text = "0" Then
        MsgBox "Footing Thickness must be bigger than '0'"
End If
End Sub

Private Sub load_assist_Click()
MsgBox "Do NOT input Factored Load." + vbCrLf + "This value will be calculated internally.",
vbInformation, "Information"
End Sub

Private Sub Main_Quantity_Change()
Result_Quantity.Text = Main_Quantity.Text
Result_quantity1.Text = Main_Quantity.Text
End Sub

Private Sub Material_assist_Click()
MsgBox " Typical Steel strength = 60000 psi " + vbCrLf + " Typical Conc. Strength = 4000 or 3000
psi ", vbInformation, "Typical Material Strength"
End Sub

Private Sub RC_Icon_Click()
PIC_ALL.Visible = False
PIC_Clear.Visible = True
PIC_Detail.Visible = False
End Sub

Private Sub ReturnSteel_Click()
Foot_Tab.Tab = 2
End Sub

Private Sub ReturnThick_Click()
Foot_Tab.Tab = 1
End Sub
Private Sub ReturnWidth_Click()
Foot_Tab.Tab = 1
```

```
End Sub

Private Sub Shape_Icon_Click()
PIC_ALL.Visible = True
PIC_Clear.Visible = False
PIC_Detail.Visible = False
End Sub

Private Sub Size_Assist_Click()
MsgBox "This table provides Possible Footing Sizes to resist Punching and Beam shear.",
vbInformation, "Information"
End Sub

Private Sub Soil_assist_Click()
MsgBox "This program does NOT use Ultimate Soil Pressure for footing design" + vbCrLf + "" +
vbCrLf + "Typical Allowable Soil Pressure" + vbCrLf + "     Poor soil      : 1500psf" + vbCrLf + "
Regular soil : 2000psf" + vbCrLf + "     Good soil     : 3000 - 5000psf", vbInformation, "Information"
End Sub

Private Sub step_01_Click()
Factor_Load.Text = (1.4 * Foot_DL.Text) + (1.7 * Foot_LL.Text)
Foot_List.Cls
Foot_Tk = Val(UnderDepth_Foot.Text) - 0.5
'Compute the possible footing size (DO~LOOP procedure)
Do
        'footing selfweight
        Ft_Sw = 0.15 * Foot_Tk
        'the weight of earth on the top of footing
        Soil_Wt = Wt_Earth.Text * (Val(UnderDepth_Foot.Text) - Foot_Tk) / 1000
        'the effective allowable soil pressure
        Esoil_Pres = (Soil_Pres.Text / 1000) - Ft_Sw - Soil_Wt
        'footing area
        Rfoot_Size = (Val(Foot_DL.Text) + Val(Foot_LL.Text)) / Esoil_Pres
        'one side length
        Rfoot_Width = Sqr(Rfoot_Size)
        'Rutine to make constructable footing width (ex. 8'-6" or 9")
        If Rfoot_Width > Fix(Rfoot_Width) + 0.5 Then
                Rfoot_Width = Round(Rfoot_Width)
        Else
                Rfoot_Width = Fix(Rfoot_Width) + 0.5
        End If
        'Comfirmed footing area
        Rfoot_Size = (Rfoot_Width) ^ 2
        'Pu factored soil pressure from superimposed loads
        P_Ultimate = Val(Factor_Load.Text) / Rfoot_Size
        'Effective thickness
        E_Depth = (Foot_Tk * 12) - 4
        'check two-way shear
        'Total factored shear acting on the critical section for two-way shear
        Factored_Shear = P_Ultimate * (Rfoot_Size - (((Val(Column_Thick.Text) + E_Depth) / 12)
        ^ 2))
        'The shear strength of concrete one of smallest shears
        Conc_Shear1 = (2 + 4) * (Sqr(Val(Conc_type.Text))) * (Val(Column_Thick.Text) +
        E_Depth) * 4 * E_Depth
```

```
        Conc_Shear2 = ((40 * E_Depth) / ((Val(Column_Thick.Text) + E_Depth) * 4) + 2) *
        (Sqr(Val(Conc_type.Text))) * (Val(Column_Thick.Text) + E_Depth) * 4 * E_Depth
        Conc_Shear3 = (4) * (Sqr(Val(Conc_type.Text))) * (Val(Column_Thick.Text) + E_Depth) *
        4 * E_Depth
        ' Conc_shear1 is changed by ratio of footing size
        ' in case of square footing, conc_shear1 is always bigger than conc_shear3
        ' so compare conc_shear2 and conc_shear3 only
        If Conc_Shear2 > Conc_Shear3 Then
          Conc_Shear = 0.85 * Conc_Shear3 / 1000
        Else
          Conc_Shear = 0.85 * Conc_Shear2 / 1000
        End If
        'One-way shear check
        'Total factored shear for one-way critical section
        Onefactored_Shear = P_Ultimate * Rfoot_Width * (((Rfoot_Width -
        (Val(Column_Thick.Text) / 12) - (2 * E_Depth / 12))) / 2)
        'The concrete shear strength multiplied with reduction factor 0.85
        Oneconc_Shear = 0.85 * ((2) * (Sqr(Val(Conc_type.Text))) * Rfoot_Width * 12 * E_Depth)
        / 1000
        ' Compare Factored shear and Concrete shear
        ' ONLY in case Conc_Shear > Factored_Shear and oneconc_shear> onefactored_shear (OK)
        If Conc_Shear > Factored_Shear Then
                If Oneconc_Shear > Onefactored_Shear Then
                        Foot_List.Print Foot_Tk, Format(Rfoot_Width, "####.##")
                        Foot_Thick.Text = Foot_Tk
                        WD_TXT.Text = Format(Rfoot_Width, "####.##")
                        Debug.Print Format(Rfoot_Width, "####.##")
                End If
        End If
        Foot_Tk = Foot_Tk - 0.5
Loop Until Foot_Tk < 1

'Drawing the plan view of footing and column
ScaleDraw.ScaleWidth = 2 * Rfoot_Width
ScaleDraw.ScaleHeight = 2 * Rfoot_Width * 7 / 9
ScaleDraw.ScaleTop = -(14 * Rfoot_Width / 18)
ScaleDraw.ScaleLeft = -1 * Rfoot_Width
DR_Footing.Height = Rfoot_Width
DR_Footing.Width = Rfoot_Width
DR_Column.Height = Column_Thick.Text / 12
DR_Column.Width = Column_Thick.Text / 12
DR_Footing.Top = -1 * Rfoot_Width / 2
DR_Footing.Left = -1 * Rfoot_Width / 2
DR_Column.Top = -1 * Column_Thick.Text / 24
DR_Column.Left = -1 * Column_Thick.Text / 24

'Drawing the plan view of footing and column in Tab3
Scaledraw2.ScaleWidth = 1.2 * Rfoot_Width * 12
Scaledraw2.ScaleHeight = 1.2 * Rfoot_Width * 12
Scaledraw2.ScaleTop = -0.6 * Rfoot_Width * 12
Scaledraw2.ScaleLeft = -0.6 * Rfoot_Width * 12
DR_Footing2.Height = Rfoot_Width * 12
DR_Footing2.Width = Rfoot_Width * 12
DR_Column2.Height = Column_Thick.Text
DR_Column2.Width = Column_Thick.Text
```

```
DR_Footing2.Top = -1 * Rfoot_Width * 12 / 2
DR_Footing2.Left = -1 * Rfoot_Width * 12 / 2
DR_Column2.Top = -1 * Column_Thick.Text / 2
DR_Column2.Left = -1 * Column_Thick.Text / 2

'Drawing the plan view of footing and column in Tab5
Scaledraw3.ScaleWidth = Rfoot_Width * 12
Scaledraw3.ScaleHeight = Rfoot_Width * 12
Scaledraw3.ScaleTop = -0.5 * Rfoot_Width * 12
Scaledraw3.ScaleLeft = -0.5 * Rfoot_Width * 12
DR_Column2.Height = Column_Thick.Text
DR_Column2.Width = Column_Thick.Text
DR_Column2.Top = -1 * Column_Thick.Text / 2
DR_Column2.Left = -1 * Column_Thick.Text / 2
Foot_Tab.Tab = 1
End Sub

Private Sub Step_02_Click()
'Compute the design moment
M_Design = P_Ultimate * (Val(WD_TXT.Text)) * ((((Val(WD_TXT.Text) - Val(Column_Thick.Text / 12)) / 2) ^ 2) / 2)
Design_MO.Text = Round(M_Design)

'Compute Coefficient of Resistance
K_Bar = (M_Design * 12) / (0.9 * (Val(WD_TXT.Text) * 12) * ((Val(Foot_Thick.Text) * 12 - 4) ^ 2))

'Compute the required steel ratio
Steel_R = (Val(Steel_type.Text) - Sqr((Steel_type.Text ^ 2) - (2.352 * (Steel_type.Text ^ 2) * (K_Bar * 1000) / Conc_type.Text))) / (1.176 * (Steel_type.Text ^ 2) / Val(Conc_type.Text))

'Compute the required steel area - fix( *100)/100 syntax
Req_Steel.Text = Fix((Steel_R * (Val(WD_TXT.Text) * 12) * (Val(Foot_Thick.Text) * 12 - 4)) * 100) / 100

'Compute the minimum steel ratio
If 3 * Sqr(Conc_type.Text) >= 200 Then
        Steel_Rmin = (3 * Sqr(Conc_type.Text) / Steel_type.Text)
Else
        Steel_Rmin = (200 / Steel_type.Text)
End If

'Compute the minimum steel area - fix( *100)/100 syntax
Min_Steel.Text = Fix((Steel_Rmin * (Val(WD_TXT.Text) * 12) * (Val(Foot_Thick.Text) * 12 - 4)) * 100) / 100

'Compute the temperature steel area - fix( *100)/100 syntax
Temp_Steel.Text = Fix((0.0018 * (Val(WD_TXT.Text) * 12) * (Val(Foot_Thick.Text) * 12 - 4)) * 100) / 100

'Decide the design steel area.
If Val(Req_Steel.Text) > Val(Min_Steel.Text) Then
        Req_steel1.Text = Val(Req_Steel.Text)
Else
        Req_steel1.Text = Val(Min_Steel.Text)
```

```
End If

Foot_Tab.Tab = 2
End Sub

Private Sub Step_03_Click()
'the bars are not top bars
Alpha = 1

'The bars are uncoated - black steel
Beta = 1

If Mainbar_Size < 0.44 Then
        Delta = 0.8
Else
        Delta = 1
End If

'Normal weight concrete
Gamma = 1

'Determine C value
If (3 + SteelDia / 2) > (Val(WD_TXT.Text) * 12 - (2 * (3 + SteelDia / 2)) / 24) Then
        C_Factor = (Val(WD_TXT.Text) * 12 - (2 * (3 + SteelDia / 2)) / 24)
Else
        C_Factor = (3 + SteelDia / 2)
End If

'Check (c+Ktr)/db<=2.5
If (C_Factor / SteelDia) > 2.5 Then
        C_db = 2.5
Else
        C_db = (C_Factor / SteelDia)
End If

'Compute the excess reinforcement factor
K_ER = Val(Req_steel1.Text) / (Mainbar_Size * Main_Quantity)

'Developement Length
L_Develope = ((3 * Steel_type.Text) / (40 * Sqr(Conc_type.Text))) * ((Alpha * Beta * Gamma *
Delta) / (C_db)) * (SteelDia) * K_ER
R_D_Length.Text = Fix(L_Develope * 100) / 100
P_D_Length.Text = ((Val(WD_TXT.Text) * 12 - Val(Column_Thick.Text)) / 2 - 3)
P_D_Length1.Text = P_D_Length.Text
If R_D_Length.Text > P_D_Length.Text Then
        Bad_Develope.Visible = True
        Good_Develope.Visible = False
Else
        Bad_Develope.Visible = False
        Good_Develope.Visible = True
End If
Foot_Tab.Tab = 3
End Sub
Private Sub Step_04_Click()
Foot_Tab.Tab = 4
```

```
End Sub

Private Sub UnderDepth_Foot_Change()
If UnderDepth_Foot.Text > 19 Then
        MsgBox "Depth is too deep!"
        UnderDepth_Foot.SelStart = 0
        UnderDepth_Foot.SelLength = 5
End If
End Sub

Private Sub WD_TXT_Change()
' for prevent type mismatch error by null inputing
If WD_TXT.Text = "" Then
        WD_TXT.Text = Rfoot_Width
        WD_TXT.SelStart = 0
        WD_TXT.SelLength = 5
End If

Rfoot_Width = Val(WD_TXT.Text)
ScaleDraw.ScaleWidth = 2 * Rfoot_Width
ScaleDraw.ScaleHeight = 2 * Rfoot_Width * 7 / 9
ScaleDraw.ScaleTop = -(14 * Rfoot_Width / 18)
ScaleDraw.ScaleLeft = -1 * Rfoot_Width
DR_Footing.Height = Rfoot_Width
DR_Footing.Width = Rfoot_Width
DR_Column.Height = Column_Thick.Text / 12
DR_Column.Width = Column_Thick.Text / 12
DR_Footing.Top = -1 * Rfoot_Width / 2
DR_Footing.Left = -1 * Rfoot_Width / 2
DR_Column.Top = -1 * Column_Thick.Text / 24
DR_Column.Left = -1 * Column_Thick.Text / 24

'Drawing the plan view of footing and column in Tab3
Scaledraw2.ScaleWidth = 1.2 * Rfoot_Width * 12
Scaledraw2.ScaleHeight = 1.2 * Rfoot_Width * 12
Scaledraw2.ScaleTop = -0.6 * Rfoot_Width * 12
Scaledraw2.ScaleLeft = -0.6 * Rfoot_Width * 12
DR_Footing2.Height = Rfoot_Width * 12
DR_Footing2.Width = Rfoot_Width * 12
DR_Column2.Height = Column_Thick.Text
DR_Column2.Width = Column_Thick.Text
DR_Footing2.Top = -1 * Rfoot_Width * 12 / 2
DR_Footing2.Left = -1 * Rfoot_Width * 12 / 2
DR_Column2.Top = -1 * Column_Thick.Text / 2
DR_Column2.Left = -1 * Column_Thick.Text / 2
Result_Dim1.Text = WD_TXT.Text
Result_Dim2.Text = WD_TXT.Text
End Sub
```

## *6. Wall Footing Module*

```
' <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
' Reinforced Concrete Structure Designer (RCSD)
' <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
' WALL FOOTING MODULE
' <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
' University of Southern California, School of Architecture, Master of Building Science
' Copyright 2002 by the University of Southern California and Kang-Kyu Choi
' All right reserved
' Contact Author: Kang-Kyu Choi < hydrofall@hotmail.com >
' RCSD is a reinforced concrete structure design assistant tool for learner.
' <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
Option Explicit
Dim Ft_Sw, Soil_Wt, Esoil_Pres, Rsoil_Pres, Rfoot_Width, Foot_Tk, spacing, CaseBar1 As Double
Dim E_Depth, V_Ultimate, V_C, DSoil_Pres, M_Ultimate, K_Bar, Steel_R, Mainbar_Size,
Tempbar_Size, Max_Spacing As Double
Dim Quantity_Bar, Alpha, Beta, Delta, Gamma, C_Factor, C_db, Casebar, L_Develope As Double

Private Sub Draw_update_Click()
ScaleDraw.ScaleWidth = 2 * Rfoot_Width
ScaleDraw.ScaleHeight = 2 * Rfoot_Width * 7 / 9
ScaleDraw.ScaleTop = -(14 * Rfoot_Width / 36)
ScaleDraw.ScaleLeft = -1 * Rfoot_Width
G_line.Y1 = 0
G_line.Y2 = 0
Wall_Line.Y1 = (UnderDepth_Foot.Text - Foot_Thick.Text)
Wall_Line.BorderWidth = Rfoot_Width
DR_Footing.Height = Foot_Thick.Text
DR_Footing.Width = Rfoot_Width
DR_Footing.Top = (UnderDepth_Foot.Text - Foot_Thick.Text)
DR_Footing.Left = -(Rfoot_Width / 2)
End Sub

Private Sub barnum_Click()
Casebar = Barnum.ListIndex
Select Case Casebar
        Case 0:
                Mainbar_Size = 0.11
        Case 1:
                Mainbar_Size = 0.2
        Case 2:
                Mainbar_Size = 0.31
        Case 3:
                Mainbar_Size = 0.44
        Case 4:
                Mainbar_Size = 0.6
        Case 5:
                Mainbar_Size = 0.79
        Case 6:
                Mainbar_Size = 1
        Case 7:
                Mainbar_Size = 1.27
        Case 8:
```

```
            Mainbar_Size = 1.56
      Case 9:
            Mainbar_Size = 2.25
      Case 10:
            Mainbar_Size = 4
End Select

Main_Spacing.Text = Fix(Fix(Mainbar_Size * 100 / Val(Req_Steel.Text)) * 12 / 100)
'For Max Spacing consideration
'Max_Spacing = Mainbar_Size * Steel_type.Text / (50 * WD_TXT.Text)
Main_steeltype.Text = Barnum.Text
Main_Spacing1.Text = Main_Spacing.Text

If Main_Spacing < 6 Then
      MsgBox (" Steel spacing is too dense. Choose bigger steel!")
End If
End Sub

Private Sub BarNum1_click()
CaseBar1 = BarNum1.ListIndex
Select Case CaseBar1
      Case 0:
            Tempbar_Size = 0.11
      Case 1:
            Tempbar_Size = 0.2
      Case 2:
            Tempbar_Size = 0.31
      Case 3:
            Tempbar_Size = 0.44
      Case 4:
            Tempbar_Size = 0.6
      Case 5:
            Tempbar_Size = 0.79
      Case 6:
            Tempbar_Size = 1
      Case 7:
            Tempbar_Size = 1.27
      Case 8:
            Tempbar_Size = 1.56
      Case 9:
            Tempbar_Size = 2.25
      Case 10:
            Tempbar_Size = 4
End Select
Quantity_Bar = Val(Temp_Steel.Text) * Val(WD_TXT.Text) / Tempbar_Size
Quantity_TEMPBar.Text = Round(Quantity_Bar)
Quantity_TEMPBar1.Text = Quantity_TEMPBar.Text
If Val(Quantity_TEMPBar.Text) < 4 Then
   MsgBox ("Choose smaller steel size!")
End If
Temp_steeltype.Text = BarNum1.Text
End Sub

Private Sub Cutted_Icon_Click()
Mainbar_footing.Visible = False
```

```
Tempbar_footing.Visible = False
Cutted_footing.Visible = True
Size_Footing.Visible = False
End Sub

Private Sub Foot_Thick_Change()
' for prevent type mismatch error by null inputing
If Foot_Thick.Text = "" Then
        Foot_Thick.Text = 0
        Foot_Thick.SelStart = 0
        Foot_Thick.SelLength = 5
End If

 If Val(Foot_Thick.Text) > Val(UnderDepth_Foot.Text) Then
        Foot_Thick.Text = Val(UnderDepth_Foot.Text)
        Foot_Thick.SelStart = 0
        Foot_Thick.SelLength = 5
End If

Ft_Sw = 0.15 * Foot_Thick.Text
Soil_Wt = Wt_Earth.Text * (UnderDepth_Foot.Text - Foot_Thick.Text) / 1000
Esoil_Pres = (Soil_Pres.Text / 1000) - Ft_Sw - Soil_Wt
Rsoil_Pres = (Factor_Load.Text * Esoil_Pres) / (Val(Foot_DL.Text) + Val(Foot_LL.Text))
Rfoot_Width = Factor_Load.Text / Rsoil_Pres
ScaleDraw.ScaleWidth = 2 * Rfoot_Width
ScaleDraw.ScaleHeight = 2 * Rfoot_Width * 7 / 9
ScaleDraw.ScaleTop = -(14 * Rfoot_Width / 36)
ScaleDraw.ScaleLeft = -1 * Rfoot_Width
G_line.Y1 = 0
G_line.Y2 = 0
Wall_Line.Y1 = (UnderDepth_Foot.Text - Foot_Thick.Text)
Wall_Line.BorderWidth = 10
DR_Footing.Height = Foot_Thick.Text
DR_Footing.Width = Rfoot_Width
DR_Footing.Top = (UnderDepth_Foot.Text - Foot_Thick.Text)
DR_Footing.Left = -(Rfoot_Width / 2)
TK_TXT.Text = Format(Foot_Thick.Text, "###.##")
WD_TXT.Text = Format(Rfoot_Width, "###.##")
End Sub

Private Sub Foot_Thick_LostFocus()
' warnning massage for zero thickness footing
If Foot_Thick.Text = "0" Then
        MsgBox "Footing Thickness must be bigger than '0'"
End If
End Sub

Private Sub load_assist_Click()
MsgBox "Do NOT input Factored Load." + vbCrLf + "This value will be calculated internally.",
vbInformation, "Information"
End Sub


Private Sub Mainbar_icon_Click()
Mainbar_footing.Visible = True
```

```
Tempbar_footing.Visible = False
Cutted_footing.Visible = False
Size_Footing.Visible = False
End Sub

Private Sub Material_assist_Click()
MsgBox " Typical Steel strength = 60000 psi " + vbCrLf + " Typical Conc. Strength = 4000 or 3000
psi ", vbInformation, "Typical Material Strength"
End Sub

Private Sub ReturnSteel_Click()
Foot_Tab.Tab = 2
End Sub

Private Sub ReturnThick_Click()
Foot_Tab.Tab = 1
End Sub

Private Sub ReturnWidth_Click()
Foot_Tab.Tab = 1
End Sub

Private Sub Size_Icon_Click()
Mainbar_footing.Visible = False
Tempbar_footing.Visible = False
Cutted_footing.Visible = False
Size_Footing.Visible = True
End Sub

Private Sub Soil_assist_Click()
MsgBox "This program does NOT use Ultimate Soil Pressure for footing design" + vbCrLf + "" +
vbCrLf + "Typical Allowable Soil Pressure" + vbCrLf + "     Poor soil       : 1500psf" + vbCrLf + "
Regular soil : 2000psf" + vbCrLf + "     Good soil     : 3000 - 5000psf", vbInformation, "Information"
End Sub

Private Sub step_01_Click()
Factor_Load.Text = (1.4 * Foot_DL.Text) + (1.7 * Foot_LL.Text)
Foot_List.Cls
Foot_Tk = 1
Do
        Ft_Sw = 0.15 * Foot_Tk
        Soil_Wt = Wt_Earth.Text * (Val(UnderDepth_Foot.Text) - Foot_Tk) / 1000
        Esoil_Pres = (Soil_Pres.Text / 1000) - Ft_Sw - Soil_Wt
        Rsoil_Pres = (Factor_Load.Text * Esoil_Pres) / (Val(Foot_DL.Text) + Val(Foot_LL.Text))
        Rfoot_Width = Factor_Load.Text / Rsoil_Pres
        Foot_List.Print Foot_Tk, Format(Rfoot_Width, "####.##")
        spacing = 0.5
        If UnderDepth_Foot.Text > 15 Then
                spacing = 1
        End If
        Foot_Tk = Foot_Tk + spacing
Loop Until Foot_Tk > Val(UnderDepth_Foot.Text) - 0.5 Or Rfoot_Width < 0

Ft_Sw = 0.15
Soil_Wt = Wt_Earth.Text * (UnderDepth_Foot.Text - 1) / 1000
```

```
Esoil_Pres = (Soil_Pres.Text / 1000) - Ft_Sw - Soil_Wt
Rsoil_Pres = (Factor_Load.Text * Esoil_Pres) / (Val(Foot_DL.Text) + Val(Foot_LL.Text))
Rfoot_Width = Factor_Load.Text / Rsoil_Pres
ScaleDraw.ScaleWidth = 2 * Rfoot_Width
ScaleDraw.ScaleHeight = 2 * Rfoot_Width * 7 / 9
ScaleDraw.ScaleTop = -(14 * Rfoot_Width / 36)
ScaleDraw.ScaleLeft = -1 * Rfoot_Width
G_line.Y1 = 0
G_line.Y2 = 0
Wall_Line.Y1 = (UnderDepth_Foot.Text - Foot_Thick.Text)
Wall_Line.BorderWidth = 10
DR_Footing.Height = Foot_Thick.Text
DR_Footing.Width = Rfoot_Width
DR_Footing.Top = (UnderDepth_Foot.Text - Foot_Thick.Text)
DR_Footing.Left = -(Rfoot_Width / 2)
TK_TXT.Text = Format(Foot_Thick.Text, "###.##")
WD_TXT.Text = Format(Rfoot_Width, "###.##")
Foot_Tab.Tab = 1
End Sub

Private Sub Step_02_Click()
E_Depth = (TK_TXT.Text * 12) - 3.5
DSoil_Pres = Factor_Load.Text / Rfoot_Width
V_Ultimate = (((WD_TXT.Text * 12) - Wall_Thick.Text - (2 * E_Depth)) / 24) * DSoil_Pres
V_C = 0.85 * 2 * Sqr(Conc_type.Text) * 12 * E_Depth / 1000
Vu.Text = Format(V_Ultimate, "###.##")
oVc.Text = Format(V_C, "###.##")
If V_Ultimate > V_C Then
        ShearJurge.Caption = "NOT GOOD.                    Change the footing thickness"
        ShearJurge.ForeColor = &HFF&
        ShearBad.Visible = True
        ShearOK.Visible = False
        ShearTXT.Caption = "Shear Strength on the critical section is Greater than the shear strength of concrete. Shear Reinforcement is required"
        ReturnThick.Visible = True
Else
        ShearJurge.Caption = "GOOD"
        ShearJurge.ForeColor = &HFF0000
        ShearBad.Visible = False
        ShearOK.Visible = True
        ShearTXT.Caption = " Shear Strength on the critical section is Less than the shear strength of concrete. No shear reinforcment is required"
        ReturnThick.Visible = False
End If

M_Ultimate = (DSoil_Pres / 2) * (((Rfoot_Width - (Wall_Thick / 24)) / 2) ^ 2)
K_Bar = (M_Ultimate * 12) / (0.9 * 12 * (E_Depth ^ 2))
Steel_R = (Val(Steel_type.Text) - Sqr((Steel_type.Text ^ 2) - (2.352 * (Steel_type.Text ^ 2) * (K_Bar * 1000) / Conc_type.Text))) / (1.176 * (Steel_type.Text ^ 2) / Val(Conc_type.Text))

If V_Ultimate <= V_C Then
        If 3 * Sqr(Conc_type.Text) >= 200 Then
                Min_Steel.Text = Fix(((3 * Sqr(Conc_type.Text) * 12 * E_Depth) / Steel_type.Text) * 100) / 100
        Else
```

```
            Min_Steel.Text = Fix((200 * 12 * E_Depth / Steel_type.Text) * 100) / 100
        End If

        Req_Steel.Text = Fix((Steel_R * 12 * E_Depth) * 100) / 100

        If Val(Req_Steel.Text) > Val(Min_Steel.Text) Then
            Req_steel1.Text = Req_Steel.Text
        Else
            Req_steel1.Text = Val(Min_Steel.Text)
        End If

Temp_Steel.Text = Fix((0.0018 * 12 * Foot_Thick.Text * 12) * 100) / 100
Temp_Steel1.Text = Temp_Steel.Text
End If

Foot_Tab.Tab = 2
End Sub

Private Sub Step_03_Click()
Alpha = 1 '(the bars are not top bars)
Beta = 1  '(The bars are uncoated - black steel )

If Mainbar_Size < 0.44 Then
        Delta = 0.8
Else
        Delta = 1
End If

Gamma = 1 '(Normal weight concrete)

'Determine C value
If (3 + Sqr(Mainbar_Size / 3.14)) > Val(Main_Spacing.Text) / 2 Then
        C_Factor = Val(Main_Spacing.Text) / 2
Else
        C_Factor = (3 + Sqr(Mainbar_Size / 3.14))
End If

If C_Factor / (2 * Sqr(Mainbar_Size / 3.14)) > 2.5 Then
        C_db = 2.5
Else
        C_db = C_Factor / (2 * Sqr(Mainbar_Size / 3.14))
End If

'Developement Length
L_Develope = ((3 * Steel_type.Text) / (40 * Sqr(Conc_type.Text))) * ((Alpha * Beta * Gamma *
Delta) / (C_db)) * (2 * Sqr(Mainbar_Size / 3.14)) * ((Fix((Steel_R * 12 * E_Depth) * 100) / 100) /
(Mainbar_Size * 12 / Main_Spacing.Text))
R_D_Length.Text = Fix(L_Develope * 100) / 100
P_D_Length.Text = ((Val(WD_TXT.Text) * 12 - Val(Wall_Thick.Text)) / 2 - 3)
P_D_Length1.Text = P_D_Length.Text


If R_D_Length.Text > P_D_Length.Text Then
        Bad_Develope.Visible = True
        Good_Develope.Visible = False
```

```
Else
        Bad_Develope.Visible = False
        Good_Develope.Visible = True
End If
Foot_Tab.Tab = 3
End Sub
Private Sub Step_04_Click()
Foot_Tab.Tab = 4
End Sub

Private Sub Tempbar_Icon_Click()
Mainbar_footing.Visible = False
Tempbar_footing.Visible = True
Cutted_footing.Visible = False
Size_Footing.Visible = False
End Sub

Private Sub TK_TXT_Change()
TK_TXT1.Text = TK_TXT.Text
End Sub

Private Sub UnderDepth_Foot_Change()
If UnderDepth_Foot.Text > 19 Then
        MsgBox "Depth is too deep!"
        UnderDepth_Foot.SelStart = 0
        UnderDepth_Foot.SelLength = 5
End If
End Sub

Private Sub WD_TXT_Change()
WD_TXT1.Text = WD_TXT.Text
End Sub
```