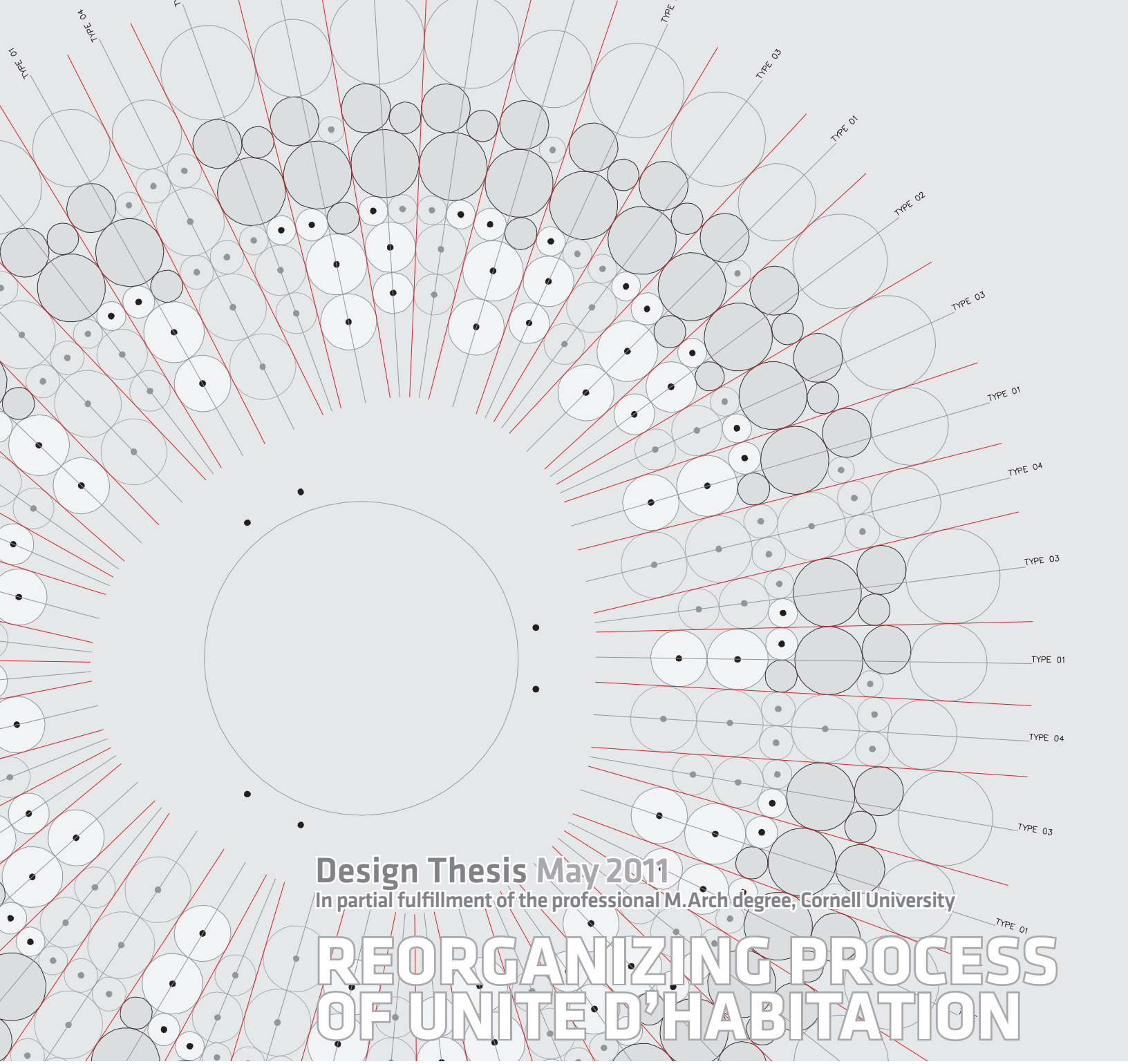


Design Thesis May 2011

In partial fulfillment of the professional M.Arch degree, Cornell University

REORGANIZING PROCESS OF UNITE D'HABITATION

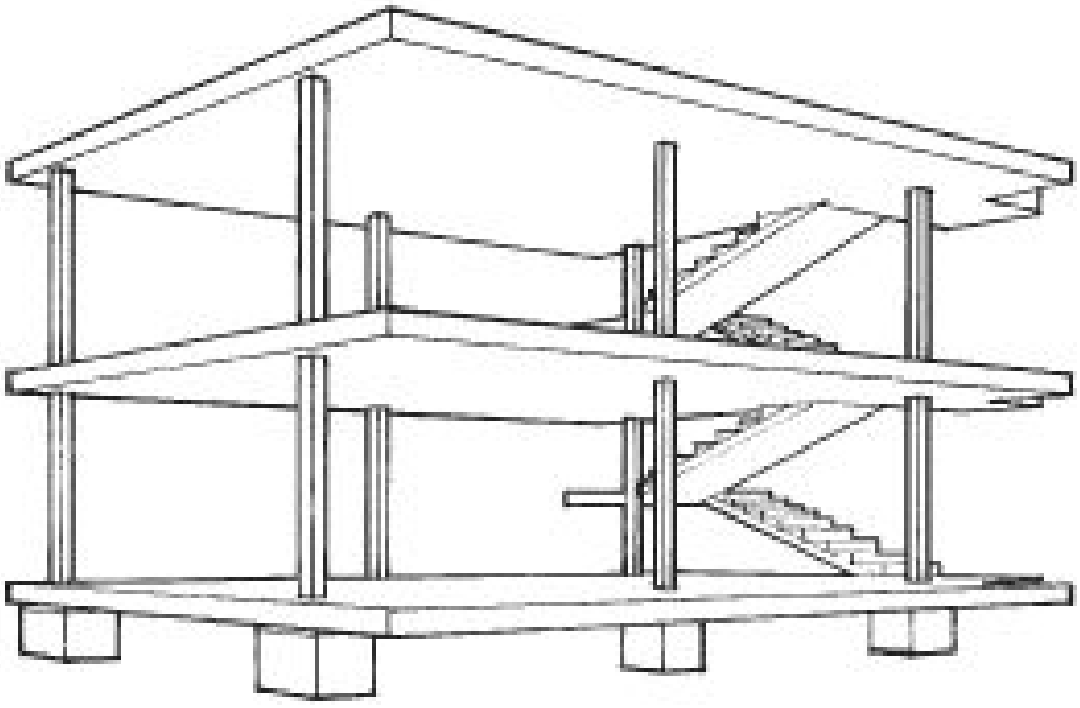
The frame developed in Chicago is to modern architecture what the column was to classic architecture. In his account, not only is the frame the catalyst of modern architecture but it has even become modern architecture, appearing when not even structurally necessary. - *from Surface Architecture by Mohsen Mostafavi*

This is the clear statement of what the frame (which is basically Domino System) means in modern architecture. The frame in modern architecture is the most hegemonic element. My thesis starts from the curiosity; what if we get rid of the frame from our modern living machines?

WOO JAE SUNG (WS92@CORNELL.EDU)
ADVISORS **ALEKSANDR MERGOLD + MICHAEL SILVER**

CONTENTS

INTRODUCTION



Five Points of Architecture by Le Corbusier

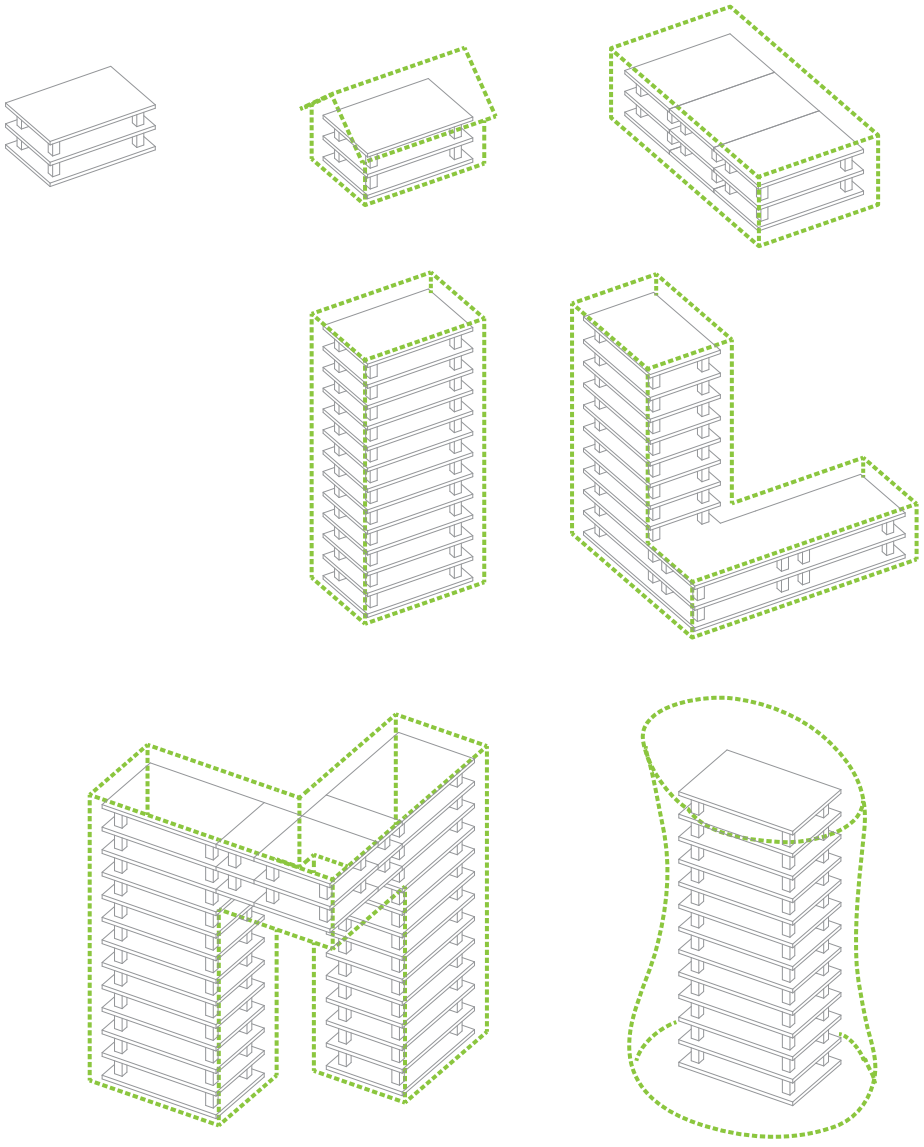
1. *The pilotis elevating the mass off the ground*
2. *The free plan, achieved through the separation of the load-bearing columns from the walls subdividing the space*
3. *The free facade, the corollary of the free plan in the vertical plane*
4. *The long horizontal sliding window*
5. *The roof garden, restoring, supposedly, the area of ground covered by the house.*

The pilotis is, needless to say, the most crucial element, because the rest of these points are more or less the result of the pilotis. However, what makes the Domino system truly the design reference for architects are the second and third items, where we are talking about design freedom. This is what I call "the Domino spirit." The Domino system is, of course, a physical platform. But its physicality should be the underlying premise for achieving flexibility within efficiency.

•••

Since the Industrial Revolution, thanks to the new technologies and materials, modern architecture had faced major changes. The most significant among them has been the advent of the Domino system. Proposed by Le Corbusier in the early 20th century, it had a far-reaching impact on modern architecture at both the physical and conceptual levels.

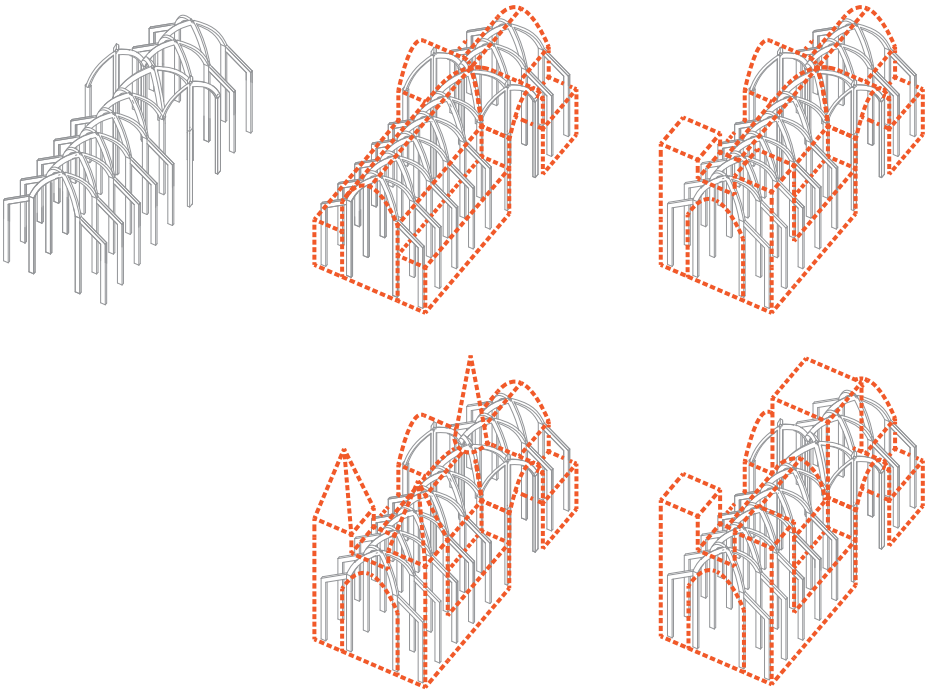
DOMINO AS A MODULAR SYSTEM



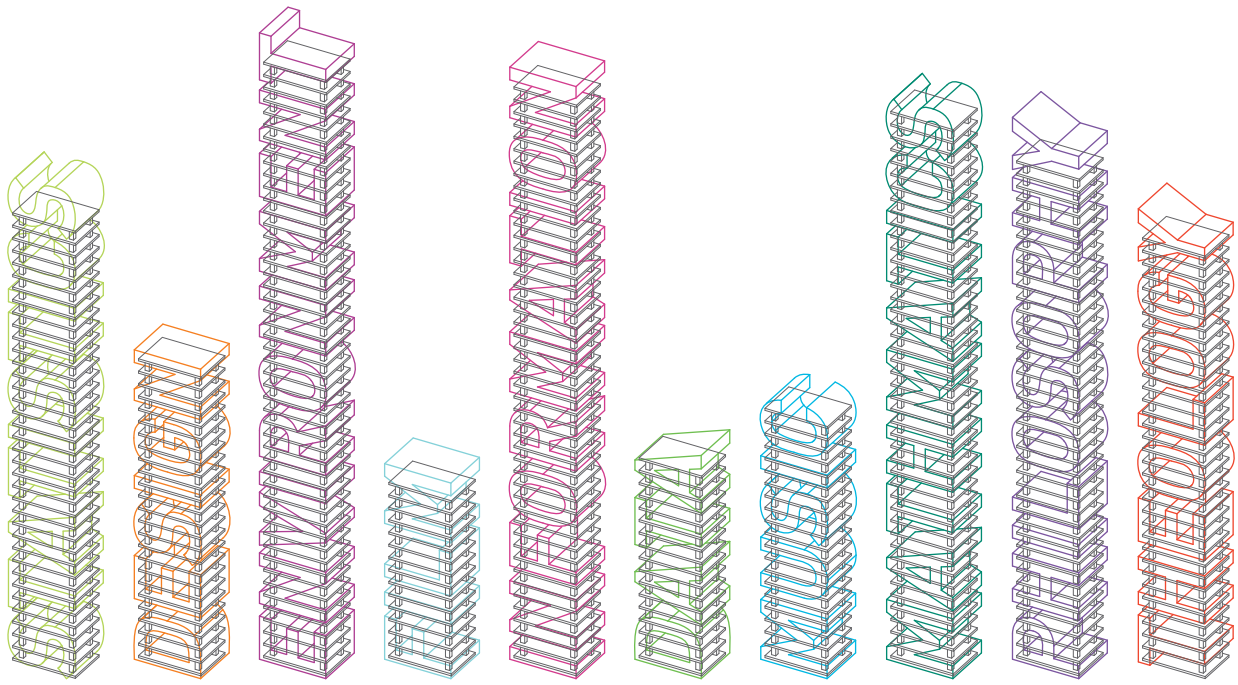
Physically, or tectonically, it re-shaped our built environment with the aid of industrial products and new means of construction. Its pure and efficient structural system, made out of reinforced concrete slabs and columns, gave architects much more freedom to design facades, plans, and the building's form itself. This created a clear distinction between the Domino system and pre-modern architectures, which were basically style-dependent[01]. Some might argue that there still exist styles in modern architecture, and this might be true. However, styles in modern architecture are not tied to tectonic issues. Rather, they are closely bonded to particular Isms. In other words, tectonic issues do not matter anymore in modern architecture thanks to the existence of the efficient structural system called Domino.

01 Surface Architecture by David Leatherbarrow and Mohsen Mostafavi

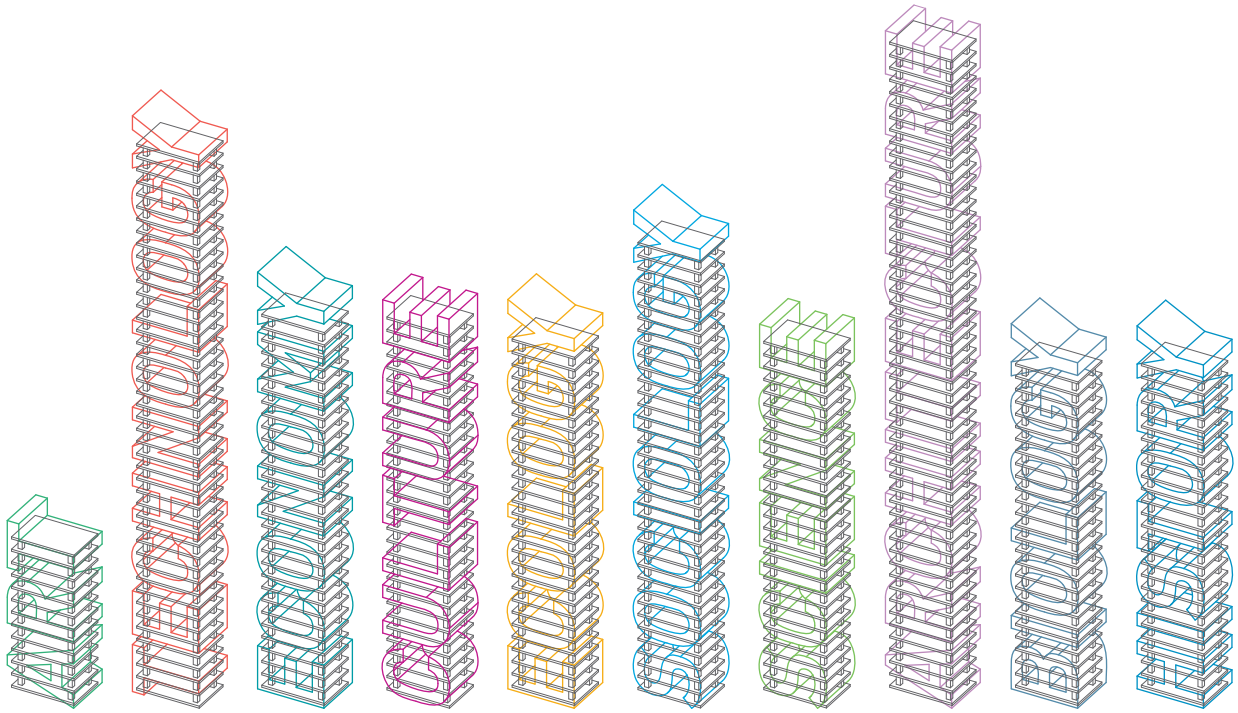
GOTHIC AS A STYLE

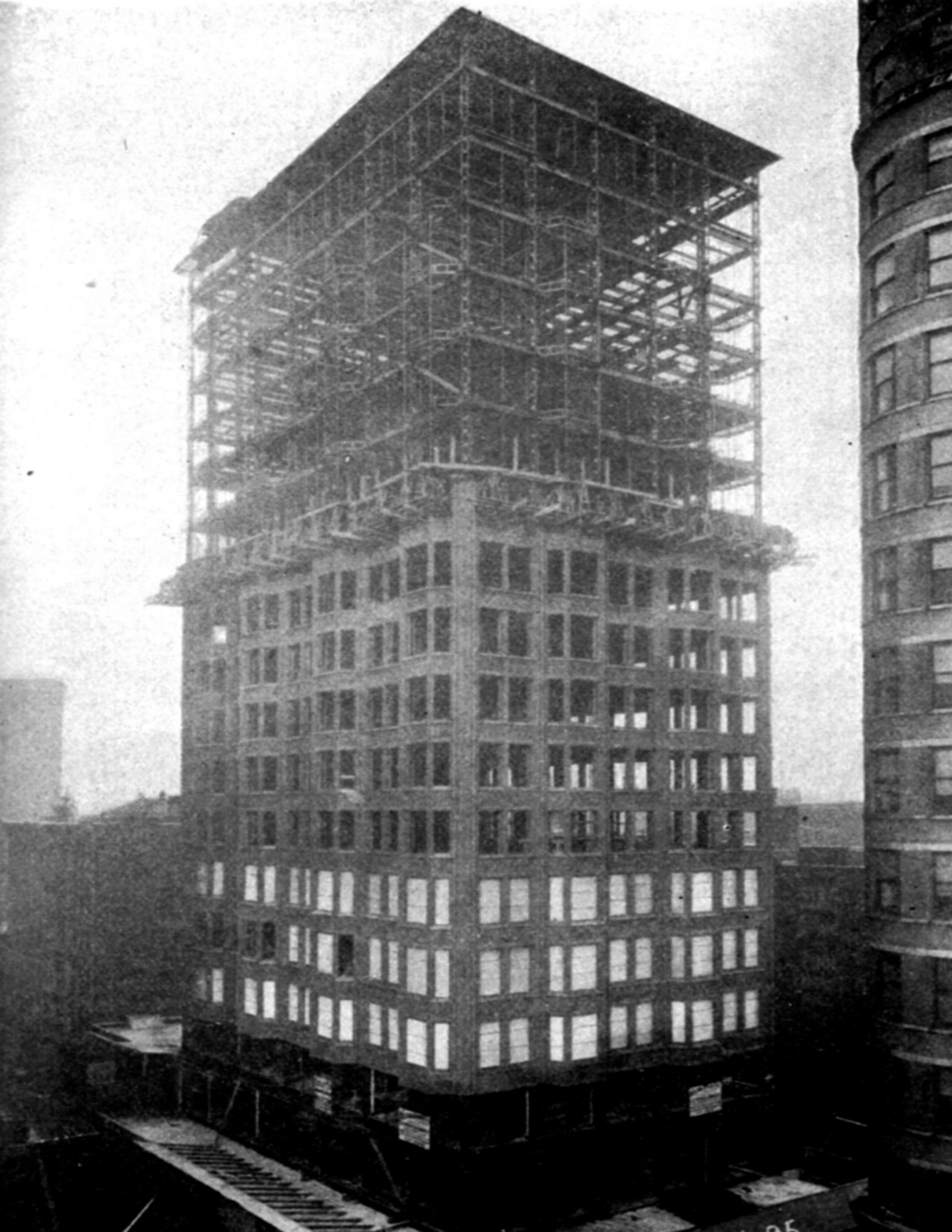


At the same time, as a reliable structural system, Domino enabled architects to broaden their design perspectives and draw inspiration from outside of architecture, adding another conceptual level on architecture. In this sense, the Domino system can be thought of as a framework by which architects can have design freedom on their work. For example, one can design a building inspired by philosophy, art, socio-cultural issues, or even by the economy, as long as the building stays stable because of Domino system.



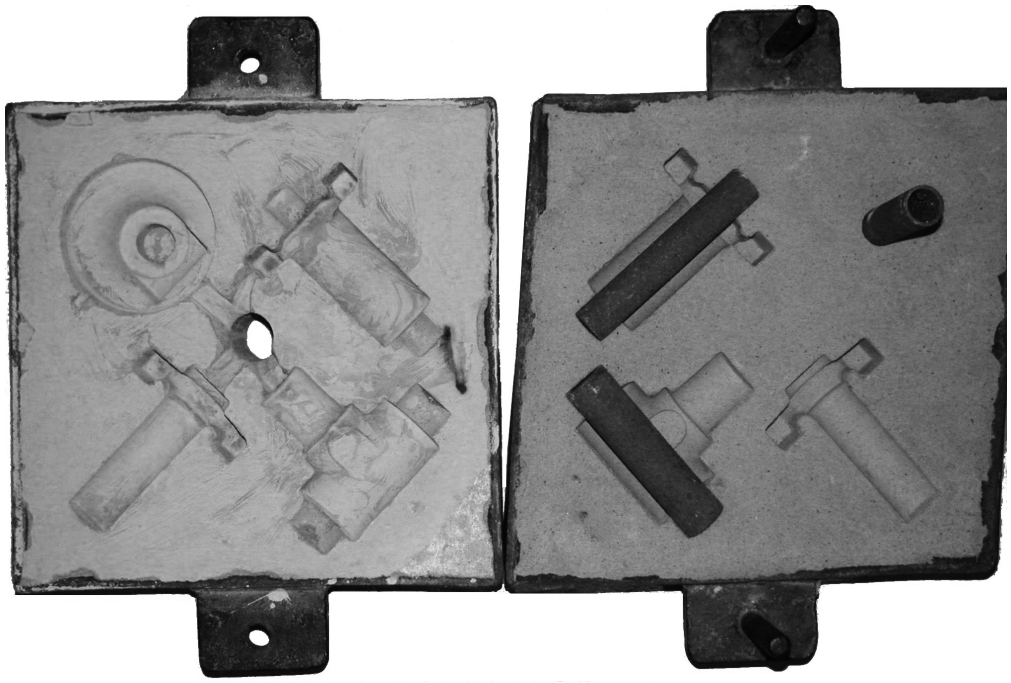
This makes it sound like the system is just a physical framework. However, the fact that it freed up modern architecture from the tectonic limitations of pre-modern architecture opened up new possibilities for interacting with the real world beyond the boundary of the discipline. This makes me hesitant to call it a mere tectonic or physical framework. Rather, it is a virtual framework. The meaning of Domino as a virtual framework can also be easily found in the Five Points of Architecture described by Le Corbusier himself; a free façade, and an open floor plan, suggesting that an architect's desire for freedom in design process can be achieved by having Domino as a fundamental base system.

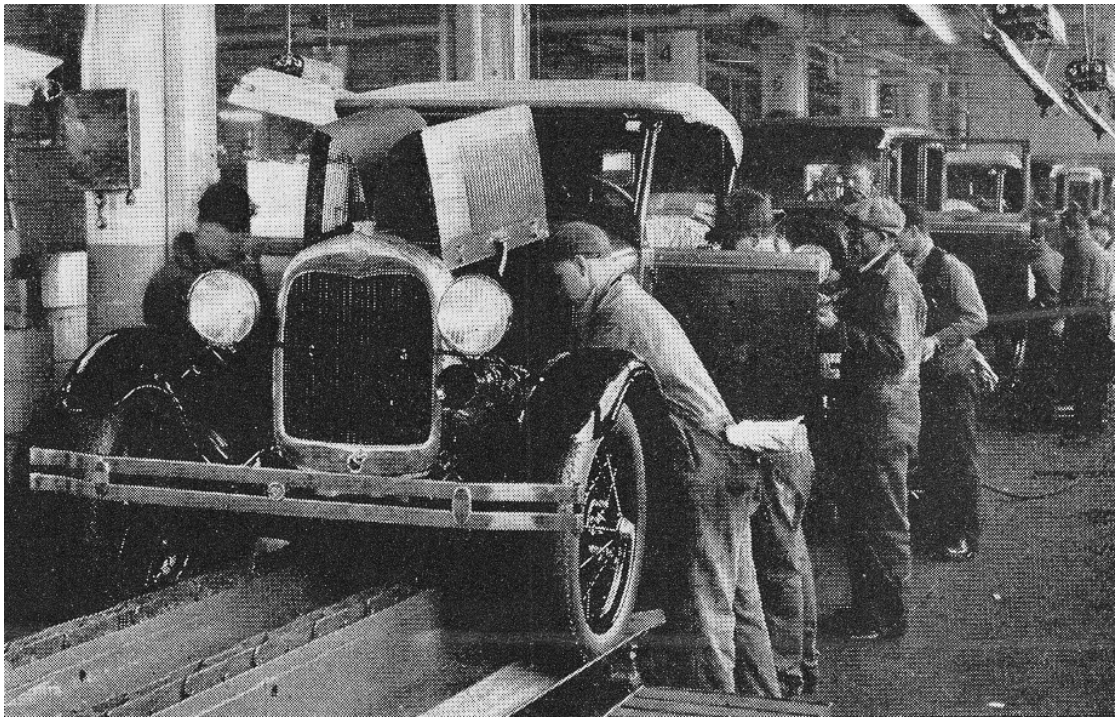




In theory, the Domino spirit, within which the system functions as a virtual framework, should work well to provide architects with much more room to play around with design issues. However, in reality, it was not as successful as expected because people abused the efficiency of the virtual framework by misinterpreting it in an economic fashion. Sadly enough, that was just the other side of the same coin; the efficient structural system that was supposed to be a sound foundation for design freedom could be also efficient in an economic sense. This captured people's attention, especially during the era of industrial revolution, allowing them to forget the design issues behind it. The system became a money-saving form-generator or platform. The Chicago frame would be a good example of this kind of abuse, although it enabled us to rethink the meaning and role of the architectural skin [01]. By the time we arrived at the era of Chicago frame, most of the buildings in cities were factory-made, which has continued to be the case up until now.

01 Surface Architecture by David Leatherbarrow and Mohsen Mostafavi
Image Left < http://architecturefarm.wordpress.com/2010/02/11/chicago-earthquake/

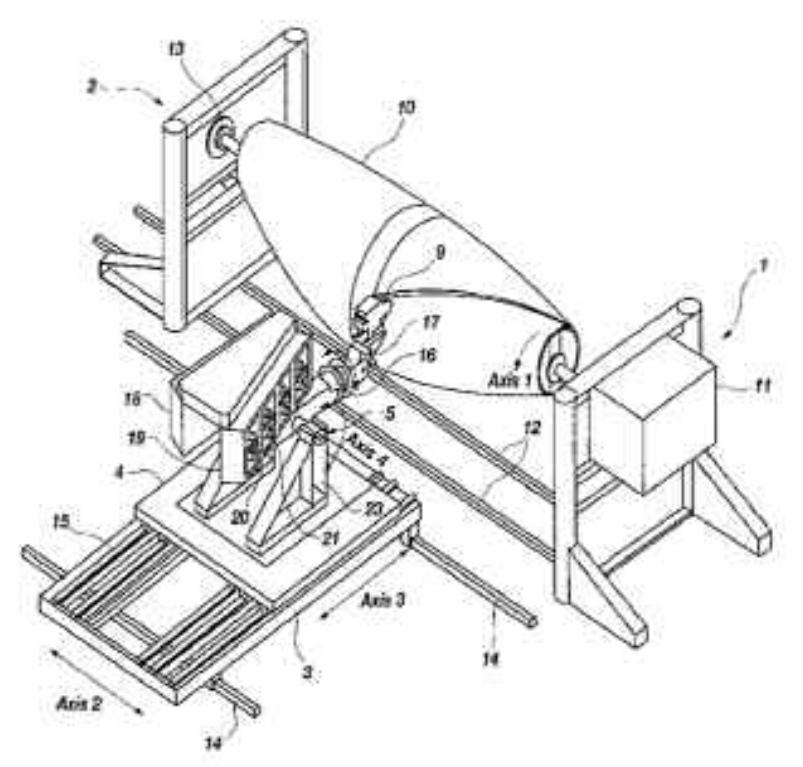




There are two major reasons that explain the gap between the ideal and reality: the social paradigm and technology that supports it. As a result of industrialization, every property in a society got capitalized and needed to be controlled financially. Under these circumstances, there was no excuse for wasting time, goods, and services. The beauty of industrialization was in the mass production system that consumed less material, time, and effort while maintaining a certain level of quality. Mass production, represented by the cast-and-mold system, in turn created a whole new aesthetic standard, represented by a functional and anti-ornamental thesis of "less is more." Of course, there were a few minor movements that exploited technology and new material to pursue more than just function and efficiency, such as Art Nouveau or the Arts and Craft Movement. However, these were just not enough to become major trends because they failed to touch the hearts of capitalists.

Image Upper Left < Cast and Mold system
Image Bottom Left < Henry Ford Model T Assembly Line

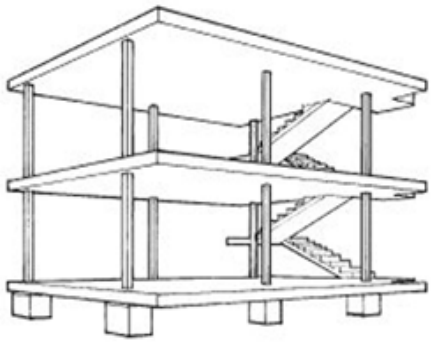

R. MUTT
1917

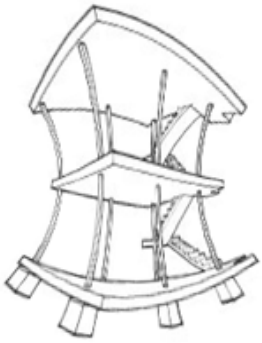

13
2
10
9
7
17
16
Axis 1
11
18
5
12
19
Axis 4
4
20
23
21
15
Axis 3
3
14
Axis 2
14

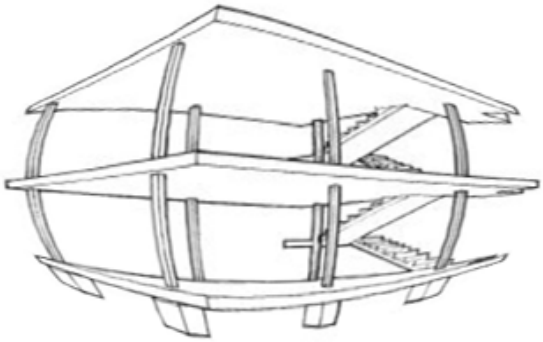
Recently, we have moved into an era of the rapid evolution where the old paradigms and systems are quickly being replaced by new ones, thanks to the advances in technology. Among these, the most significant one must be mass-customization. This century has witnessed the monotonous characteristic of mass-production, which depersonalized every single bit of our daily life in one way or another. In order to overcome this uniformity, people came up with something in between tailor-made and factory-assembled: more precisely speaking, getting tailor-made products by factory assembly systems.

An early example of what we now call the mass-customization can be found in the form of art, for example, in Marcel Duchamp's "Fountain." [01] By adding a signature on a factory-made toilet, he tried to convert the factory-made into the tailor-made. This rough idea of mass-customization evolved into the component system, which added extra steps to the existing conveyor belt system to give customers options for their needs and taste. Furthermore, radical technological improvements such as laser/water jet cutting tools, multi-axis milling machines, and fiber placement technology [02] are entirely rewriting the concept of production from "cast and mold" to "sculpting by machines." Sculpting by machine, as its name implies, does not rely on a mold, meaning that there is no limitation in customizing products. More importantly, we don't have to worry about the high costs that the customization processes used to have.

01 Image Upper Left < Marcel Duchamp's "Fountain"
02 Image Bottom Left < Vertical Studio Course Material by Michael Silver



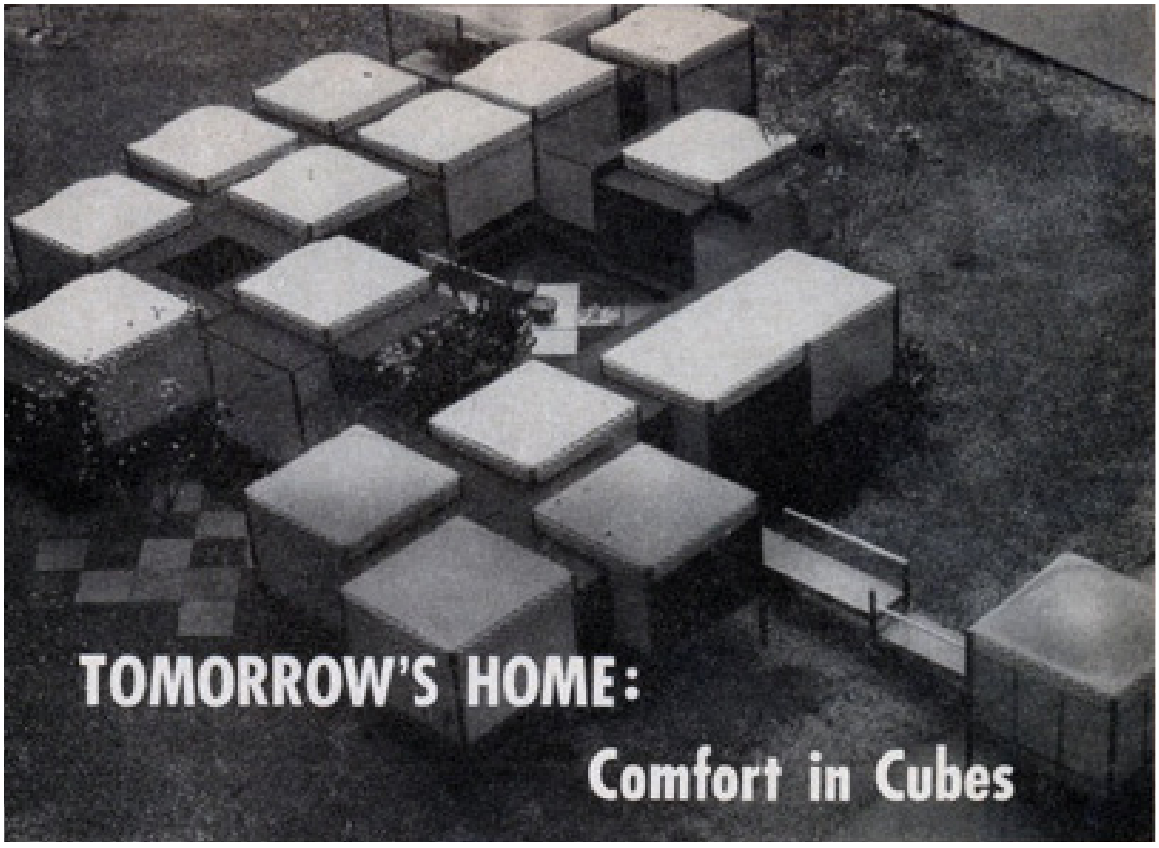




People always have new ideas, and these ideas push people to invent new technologies to support them. Sometimes, new technologies let us think differently, which can also result in new ideas being generated. Ideas and technology have been interacting with each other throughout history. Le Corbusier came up with the idea of the Domino system, inspired by industrialization and new material and technology. The system was then adopted by the capitalists of the era, modified by their standards, and has been serving as a money-saving for a long time. As mentioned above, recent advances in technology are now redefining the term "cost-effective," allowing us to reinterpret the Domino system from as a money saving form giver to an efficient virtual framework for design freedom.

With that said, in this thesis, I will propose an alternative design method based on the Domino system (or the Domino spirit), not as a money-saving, physical system, but as an efficient design reference that functions on a conceptual level. For this purpose, I will analyze a typical housing unit cluster, focusing in particular on the design and building processes, to determine the implications of the Domino system as a form giver. Next, I will theoretically remove the rigid form of the building and restore it to its intact condition, revealing only the programs and the relationships between them. Finally, I will reorganize the malleable relationship with certain internal/external logics to get a prototype of this new design method. To test the prototype, I will apply it to one of the most iconic modern living machines, Le Corbusier's Unite d'Habitation, which is in great part based on the Domino system.

PROTOTYPE

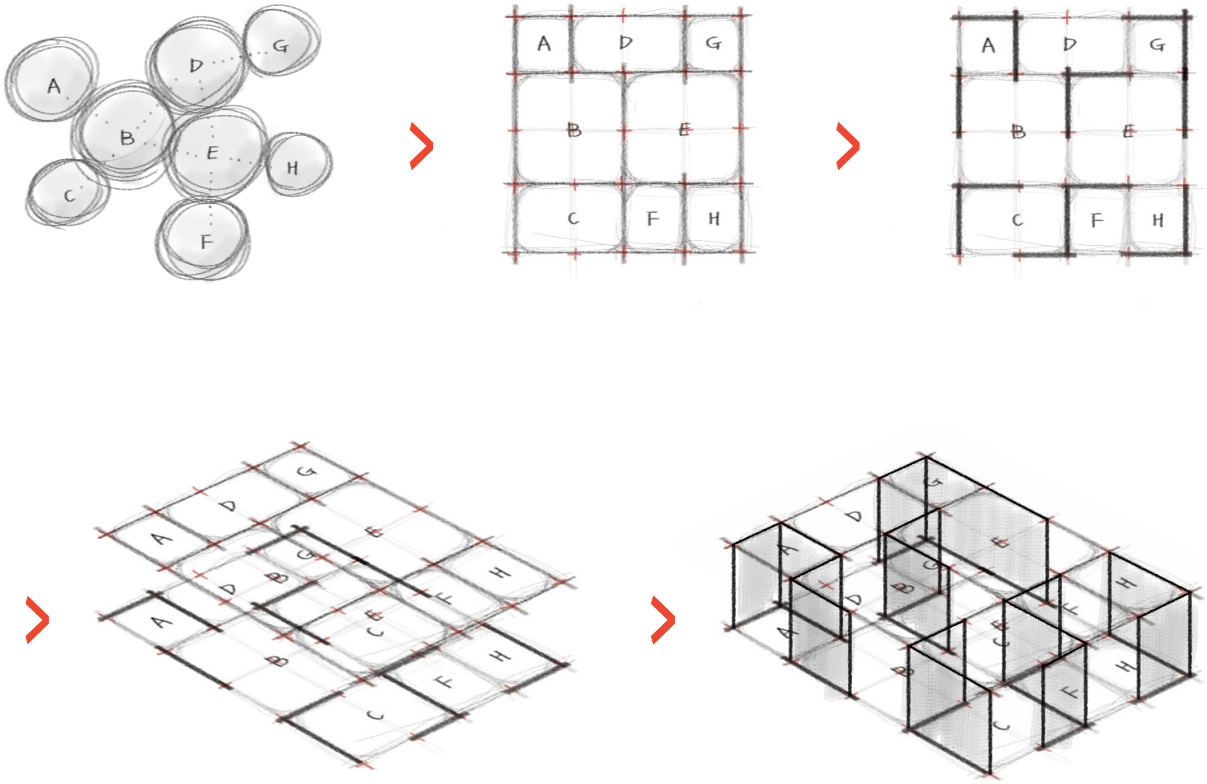

TOMORROW'S HOME:
Comfort in Cubes

Housing is one of the building types that has redefined itself since the Industrial Revolution when countless people moved into cities in search of new opportunities. Needless to say, finding affordable places to live was a main concern for many. This naturally increased the density of cities so that housing needed to be spatially efficient enough to hold as many people as possible within a limited area. Row houses or town houses[01], which we can still find in many old cities, are good examples of this type of housing. For a while, row houses were successful. Each unit followed a long and shallow plan that shared walls with the adjacent units, with couple of small windows at each end of the building. This might be an ideal system to get the maximum number of structurally stable units into a limited space with the least number of windows facing out. But the space inside was dark and cramped; due to the structural nature of the building material (bricks and mortar), the buildings could not have big enough windows to allow sufficient sunlight in, even in the daytime. Additionally, the brick structure was not strong enough for bigger/higher buildings to cover the exponentially increasing needs. In addressing this situation, the Domino system naturally captured architects' attention because it was structurally efficient as well as flexible enough to design better living environments, such as those that included big windows. For this reason, many post-industrial revolution housing types, including Unite d'Habitation, started to take advantage of the Domino system.

What was the role of the Domino system in modern housing? Was it just a physical form-giver, or a conceptual reference? As I mentioned in the previous chapter, my answer to this question is the former. It is evident when we look around our built environment that cities are full of Domino replicas.

In this prototype chapter, I will deconstruct the design/building process of typical housing units and clusters, removing all physical traces of the Domino system as a form-giver (while maintaining it as a virtual/conceptual reference system), and reorganize them to propose an alternative prototype.

01 Image Upper Left < Row House, http://www.heritageexplorer.org.uk
Image Bottom Left < Comfort in Cubes, http://prefabcosm.com

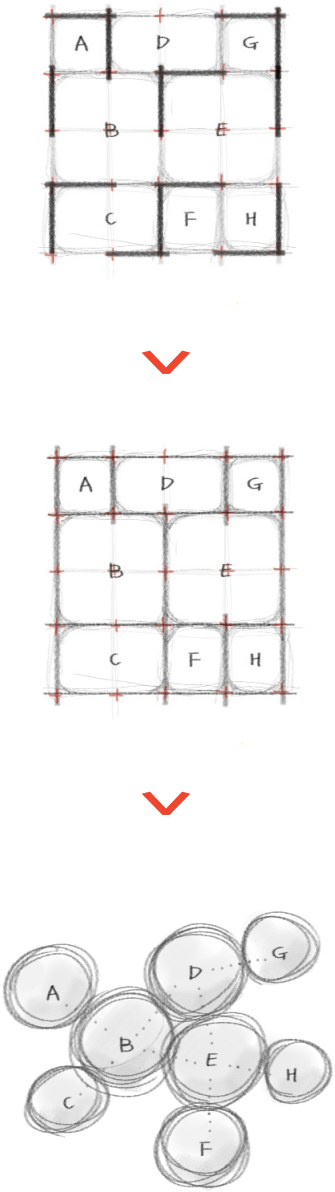


The bubble diagram is one of the most common, but still powerful tools that architects love to use. The diagram, showing bubbles of programs and their connections to each other, is in very flexible and malleable state with lots of possibilities, before it is fitted into a grid system.

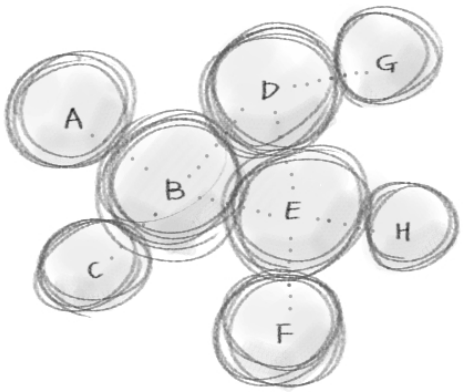
In modern architectural practice, the grid is usually calculated by economic factors such as a building's possible structural span, the number of units, or even by the dimensions of a parking spot.

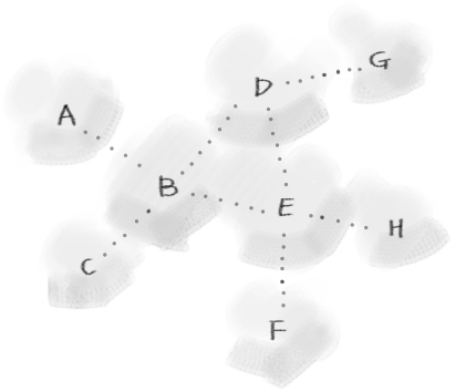
After the grid system is applied on top of the bubble diagram, two-dimensional boundaries are generated to separate or connect adjacent programs based on the bubble diagram.

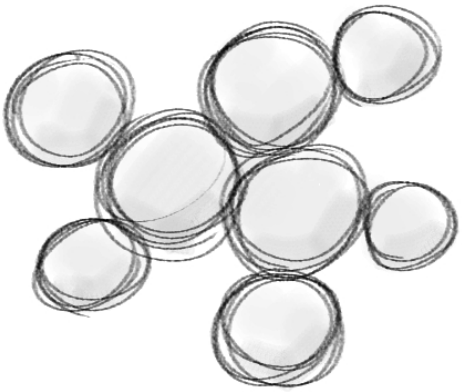
The same procedures will be repeated for every floor in the building, stacking them one by one to extrude the two-dimensional lines vertically. Extrusion is a very common, straightforward, efficient, and easy method of making walls out of two-dimensional lines.

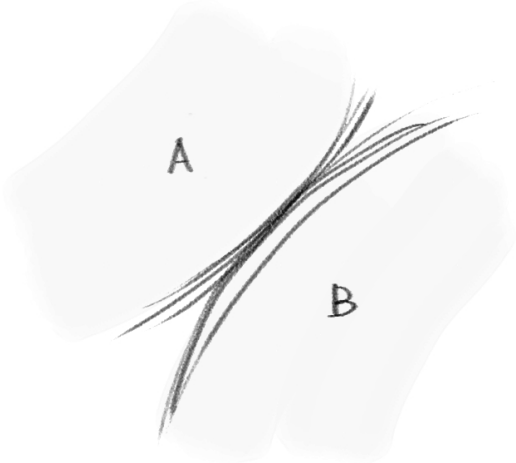


To see how the grid worked as a form-giver, remove it from the plan. But then we need an alternative way of organizing the bubbles and expanding them three dimensionally, since we don't want each bubble to float around in the air.

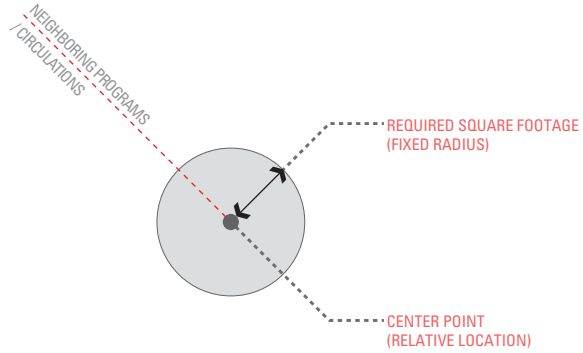




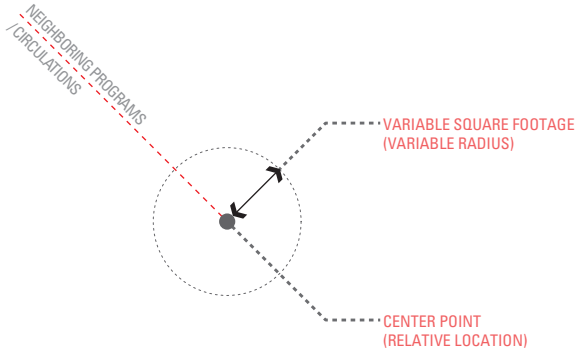




Before we discuss the packing method, we first need to have a clear definition of what the bubbles are and how they connect to each other.



CIRCLE : PROGRAM



POINT : CIRCULATION

Bubbles and Points

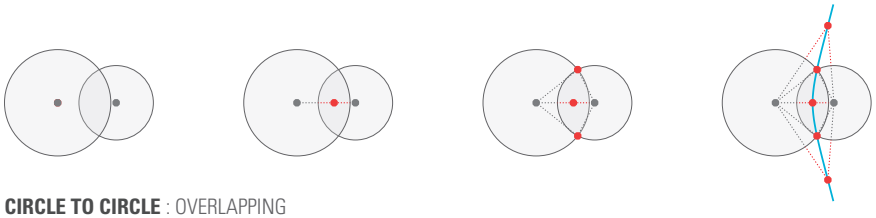
A bubble is a representation of a specific program. A program has a center point and radius to represent the required square footage.

A point without a circle around it is a representation of a corridor, a hall, or any other form of space that is dedicated to circulation.

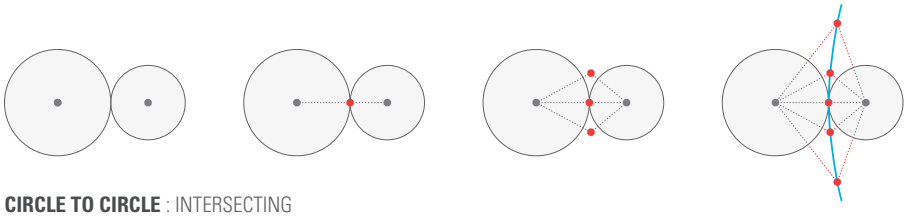
Connecting Lines

Bubbles and points can be connected by lines.
A solid line represents a physical connector such as a door, an opening, or any other form through which one can move from one space to another.

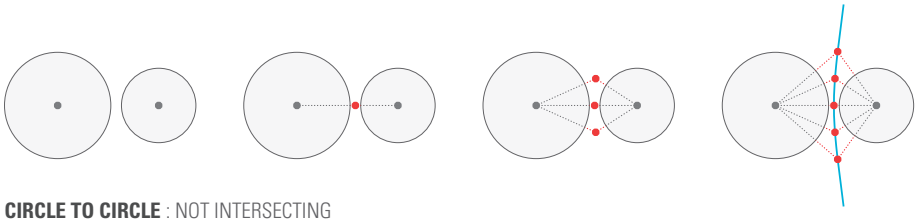
A dotted line represents visual connectors such as a partial opening, windows, or any other form that we usually do not pass through but still can see through.



CIRCLE TO CIRCLE : OVERLAPPING



CIRCLE TO CIRCLE : INTERSECTING



CIRCLE TO CIRCLE : NOT INTERSECTING

CIRCLE TO POINT : INTERSECTING

CIRCLE TO POINT : INCLUDED

CIRCLE TO POINT : EXCLUDED

POINT TO POINT : NOT INTERSECTING

How should we pack the bubbles? Based on cell-packing theory[01], there are seven possible packing scenarios that fall into three categories.

Circle to Circle

Overlapping - circles that meet at two points
Intersecting - circles that meet at a single point
Not intersecting - circles that do not meet

Circle to Point

Intersecting - a circle and a point that meet at a single point
Included - a point that is inside of a circle
Excluded - a point that is outside of a circle

Point to Point

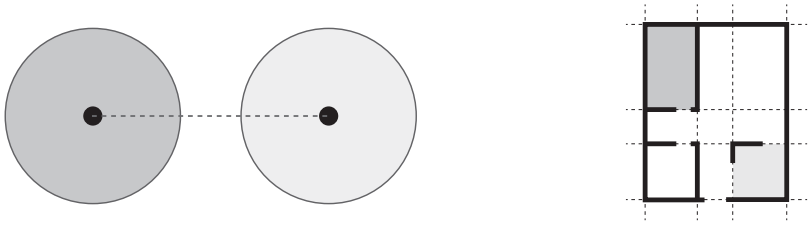
Not intersecting - points that do not meet

•••

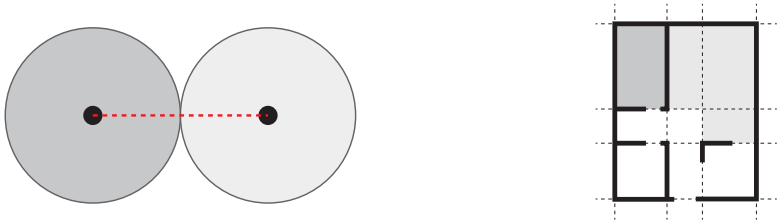
Each packing scenario generates boundaries based on the "Circle Set Voronoi / Voronoi" rule[02]. As we can see from the diagram, the boundaries generated by the rule are dependent on bubbles as well as points.

01 The Nature of Order by Christopher Alexander
02 Voronoi Diagram of a Circle Set Constructed from Voronoi Diagram of a Point Set by Deok-Soo Kim, Donguk Kim, and Kokichi Sugihara

NOT INTERSECTING PROGRAM SPACES NOT ADJACENT TO EACH OTHER



INTERSECTING PROGRAM SPACES ADJACENT TO EACH OTHER - WALL CONNECTION



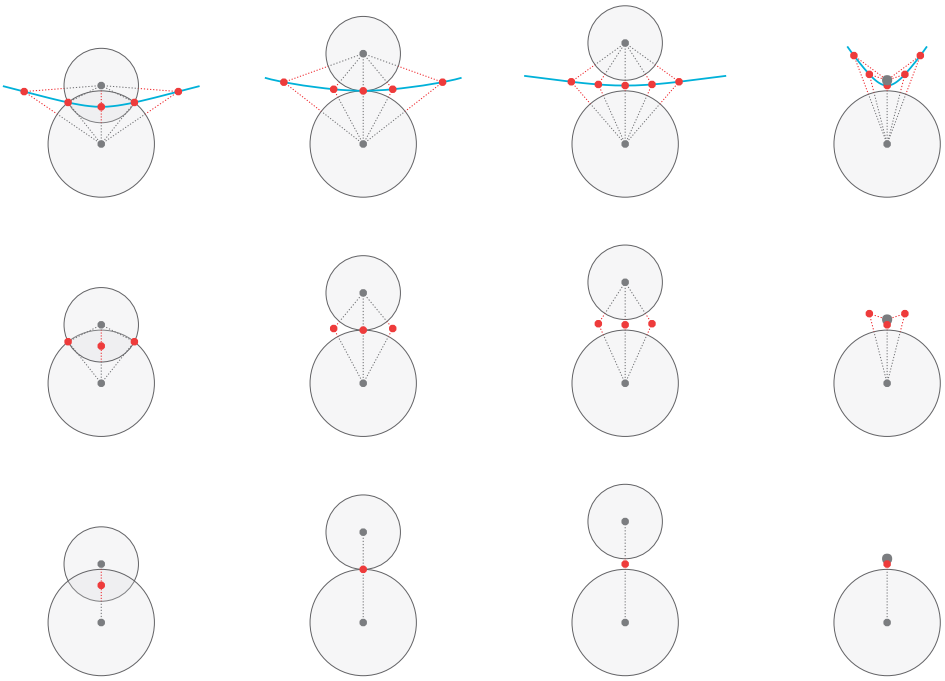
OVERLAPPING PROGRAM SPACES CONNECTED TO EACH OTHER THROUGH OPENNINGS - DOOR/OPENNING CONNECTION

CIRCULATION PROGRAM SPACE CONNECTED TO CIRCULATION PATH THROUGH DOORS / OPENNINGS

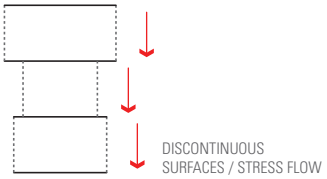
Not all of the packing examples are meaningful in real space. For example, a point-to-point connection does not necessarily have to have a boundary between points because in most cases, this type of connection implies a continuous path or a part of circulation.

By removing some packing scenarios, we can finally arrive at four meaningful packing scenarios.

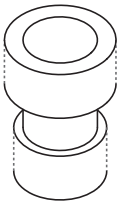
Not intersecting program spaces not adjacent to each other
Intersecting program spaces adjacent to each other / divided by wall(s)
Overlapping program spaces connected to each other through doors / openings
Circulation connection between a program space and a circulation nodal point.



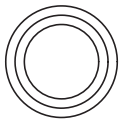
PROGRAM : DIFFERENT SIZE



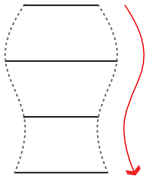
DISCONTINUOUS
SURFACES / STRESS FLOW



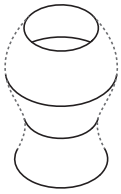
STACKING STRUCTURE



PROGRAM : DIFFERENT SIZE



CONTINUOUS
SURFACE / STRESS FLOW



PIPE CELL STRUCTURE

EXTRUDING + STACKING
DISCONTINUOUS WALLS

FORM FACTOR / VARIABLE
HEIGHT

LOFTING
CONTINUOUS WALLS

FORM FACTOR / VARIABLE
HEIGHT + RELATIVE LOCATION + BUBBLE SIZE

Unlike horizontal connections, vertical ones are simple because of the fundamental human need for flat surfaces as inhabitable spaces or platforms for our activities. If a surface has a slope and you cannot rest yourself on it for a fair amount of time, the surface is not designed for "living." It might be either for circulation, to compensate for level differences, or for "The Function of Oblique." [01]

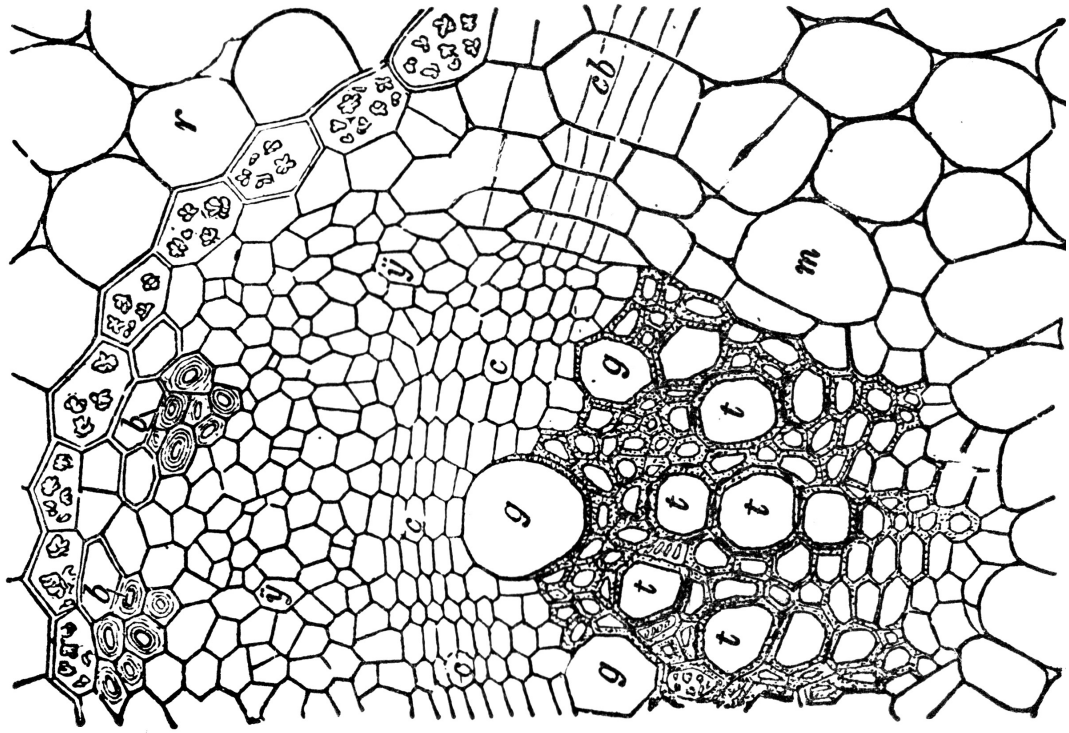
That being said, if we stack more than two program bubbles vertically and get boundaries out of them based on the cell-packing scenarios discussed in the previous pages, they will be curvilinear surfaces in section, which are not ideal geometries for holding programs. So we usually end up putting series of flat slabs in between programs stacked vertically, whatever their relationship.

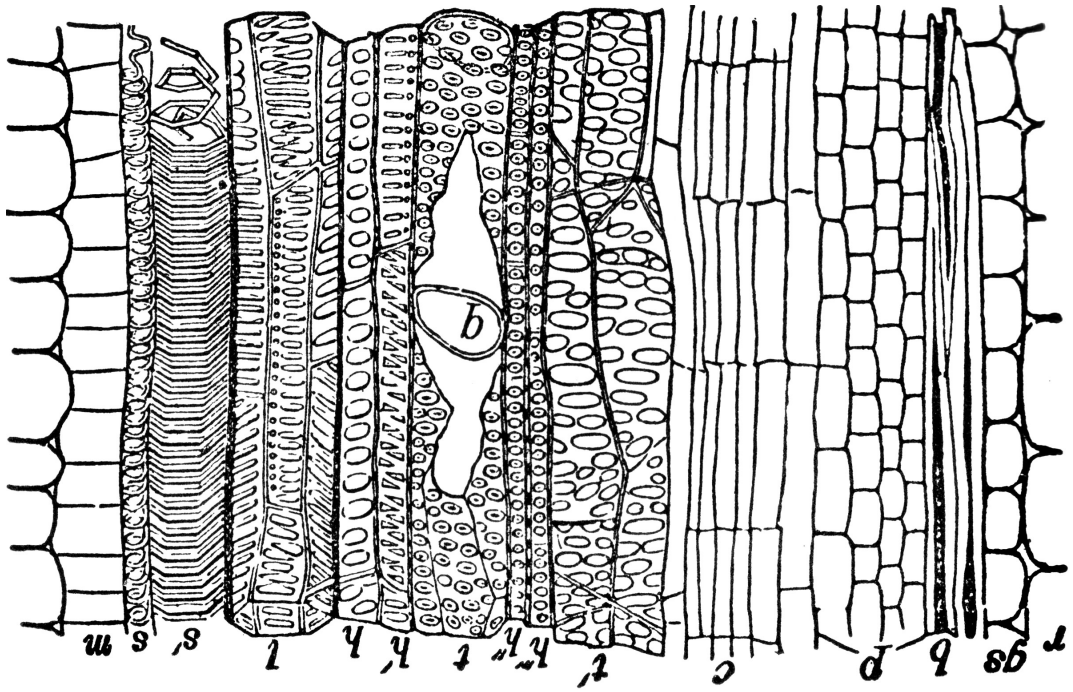
This seems to make vertical connections between (vertically stacked) program bubbles less meaningful than horizontal ones. That must be the main reason why we spend a lot of time arranging program bubbles in two-dimensional plans, then "extrude" planar boundaries vertically to get walls. Additionally, technical difficulty and cost issues on doubly curved surfaces confined architects to only using straight extrusion rather than "lofting."

Lofting, a now very familiar term to architects, is a technique for making a surface out of more than two curves. The resulting surface is dependent on the initial curves. For example, let's say we have three planar curves on different levels. By straight extrusion, the classic method of transforming two-dimensional data into three-dimensional, we can make three independent walls that barely have connections with each other. Or, we can make one continuous, streamlined wall by lofting.

A lofted surface, as its name implies, is a parameter driven by program bubbles or boundaries at different levels. Its form (wall) comes from the inner logic of a building, in this case, the relative location and size of the program bubbles. An extruded surface, on the other hand, is from a rigid system made out of something other than inner logic. In many cases from our history and today's built environment, the form factor behind it has usually been frame structures.

01 The Function of the Oblique by Paul Virilio

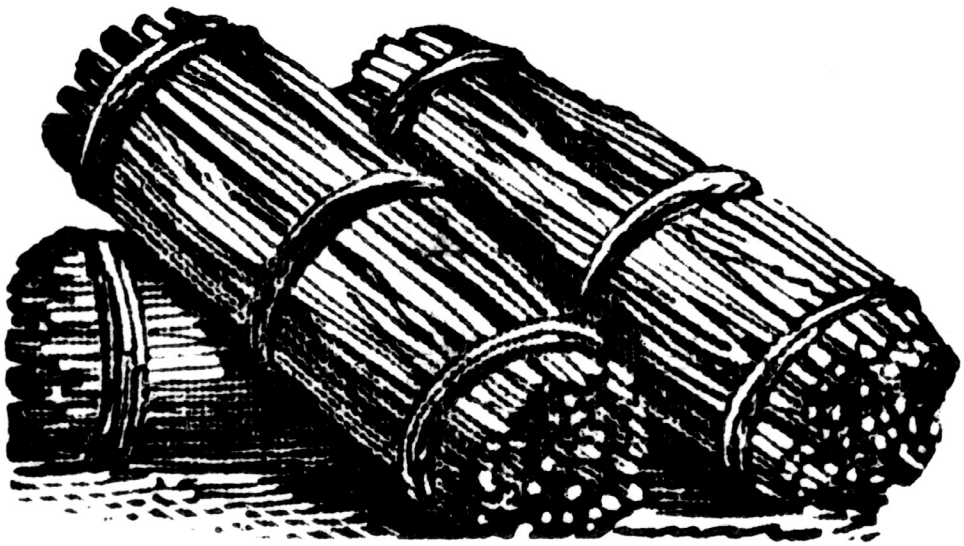

t
eb
m
ü
c
g
t
b
i


b
ш
s
s'
l
Ч
Ч'
t
Ч''
Ч'''
t'
c
p
b
gs
r

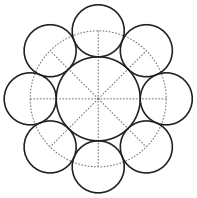
When we go back to the Five Points of Architecture by Le Corbusier, the physical characteristics of Domino gave architects the freedom to play around with plans because, unlike in row houses, the walls in the Domino system did not have to bear any loads. However, since we got rid of the rigid frame system from the traditional design method as a part of the study in alternative design processes, we need to somehow figure out how to support the system.

The single loft surface discussed in the previous pages doesn't seem to be a stable structure because it has no additional support to avoid buckling, such as flying buttresses that transfer the load all the way down to the foundations. However, when we bundle/pack individual lofted surfaces together, this makes a huge difference. The vascular bundle structure[01] of a plant serves as a good example of how this works. Although an individual vascular tube is made out of a very thin membrane that is unable even to hold its own weight, the bundled one constitutes a very efficient and powerful structural system.

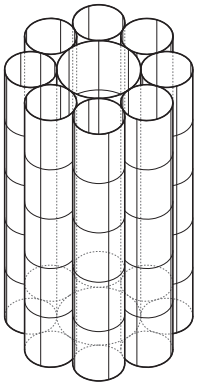
01 http://etc.usf.edu/clipart/26100/26164/fibro-vascul_26164.htm



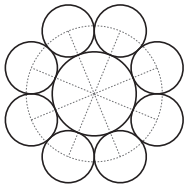
STRAIGHT EXTRUSION



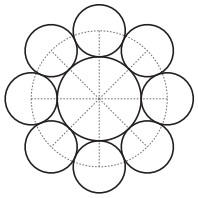
PLAN TYPE A



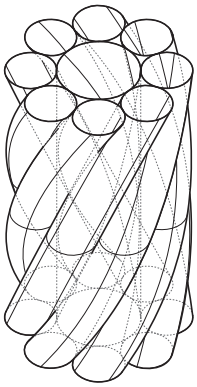
TWIST



PLAN TYPE A



PLAN TYPE A' (ROTATED)

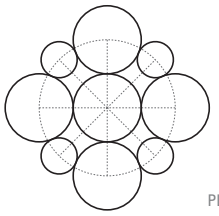


PLAN TYPE A

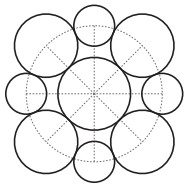
PLAN TYPE A'

PLAN TYPE A

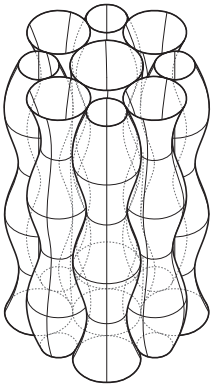
ZIGZAG



PLAN TYPE A



PLAN TYPE A



PLAN TYPE B

PLAN TYPE A

PLAN TYPE B

PLAN TYPE A

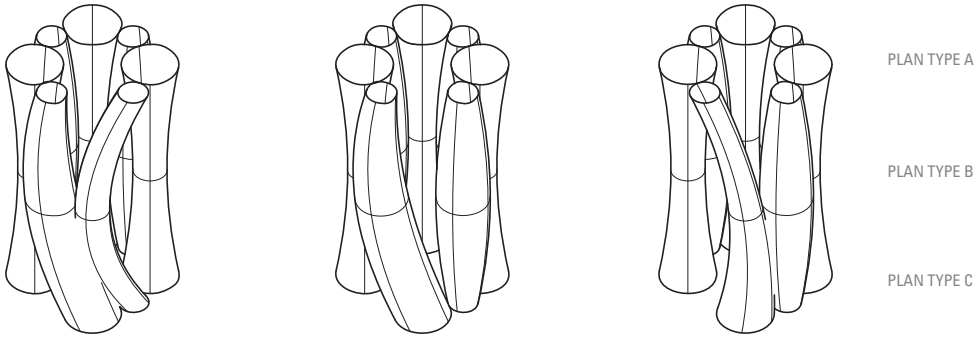
PLAN TYPE B

PLAN TYPE A

Although there might be many different ways to bundle pipes, they share one simple principle: the stability of the bundle structure is closely related to the geometric configuration of individual pipes. Whatever the geometric configuration, it must increase friction between individual components of the bundle. The easiest way to do that might be to increase the facing area to achieve maximum surface friction areas. That is how the bundle structure achieves stability. We can instinctively imagine that the straight bundle in the illustration will bond less strongly than the other two because its structure does not share as much surface area.

CASE 1 : DIFFERENT NUMBER OF PROGRAM BUBBLES

BUNDLING OPTIONS

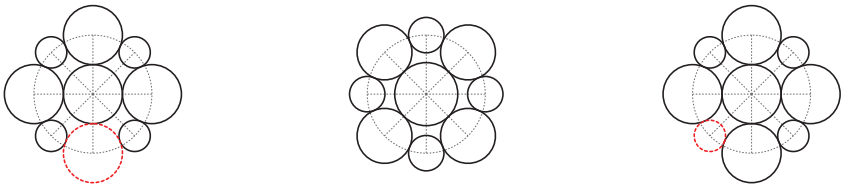


PLAN TYPE A

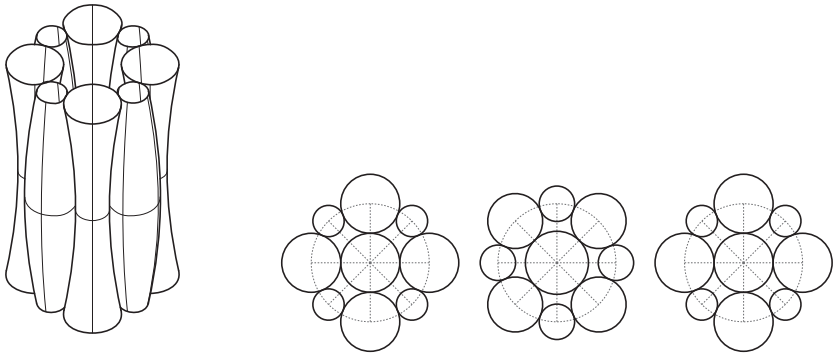
PLAN TYPE B

PLAN TYPE C

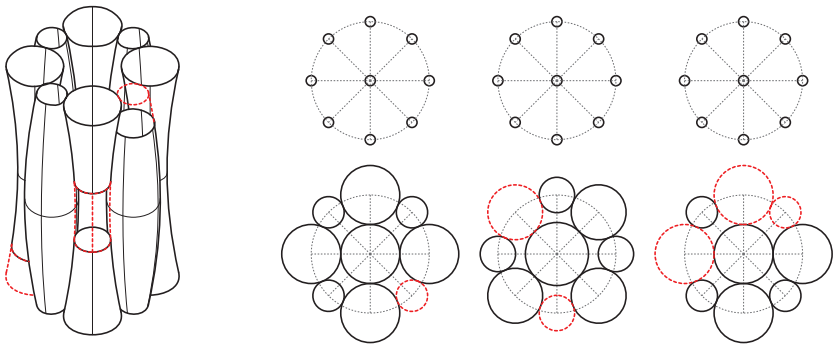
PLAN TYPES (A / B / C)



CASE 2 : SAME NUMBER OF PROGRAM BUBBLES



CASE 3 : SAME NUMBER OF PROGRAM BUBBLES WITH DUMMIES



For the bundling process, we need to consider the number of program bubbles in each layer, because lofting is basically a parametric function of at least two curves. This idea gives us two different bundling types; the first type has a different number of program bubbles in each layer and the second has the exact same number of bubbles in each layer.

Case 1 : different number of program bubbles

As I briefly mentioned above, when it comes to lofting, having different numbers of bubbles is problematic. We might need to decide which bubble we should or should not remove to match the number of bubbles in different layers. This might sound okay, but it is actually not because it constitutes a critical decision about the entire form of the building. The selection / screening process for bubbles in this instance inevitably needs to be done in a random fashion, which means less or no control over the shape of building we are designing.

Case 2 : same number of program bubbles

Since we don't have to rely on a random process, we have a perfect control in our design; the overall shape of the building is the product of our input on matters such as the number of programs, the size of the bubbles, and the relationships amongst bubbles.

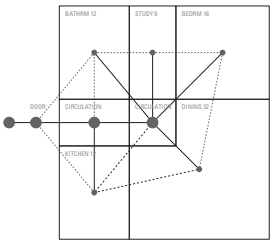
However, in reality, it seems almost impossible to have a series of floor plans with exact same number of program bubbles. Then how can we compensate for the discrepancies so as to convert this incomplete status into something complete? The answer to this question would be to use dummy cells.

Case 3 : same number of program bubbles with dummies

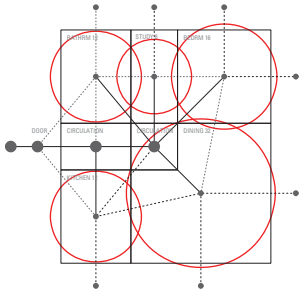
The entire body of program bubbles expanded three-dimensionally is a virtual framework that substitutes for the physical grid system of Domino. More precisely speaking, three-dimensional point grids (points that are the center points of bubbles) will be the virtual framework, and each program bubble with a specific radius based on area requirements will sit on top of the point grid shaping building plans. Needless to say, the dummy in this system will have a variable radius, depending on the size of surrounding programs, in order to accommodate a smooth transition within and between layers.

PROTOTYPING PROCESS

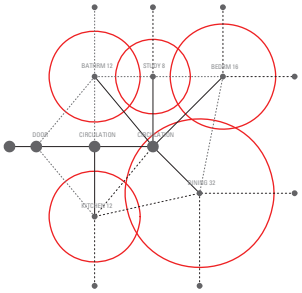
Below is the diagram of the overall prototyping process.



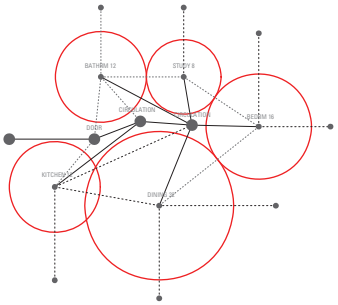
01 BASE PLAN



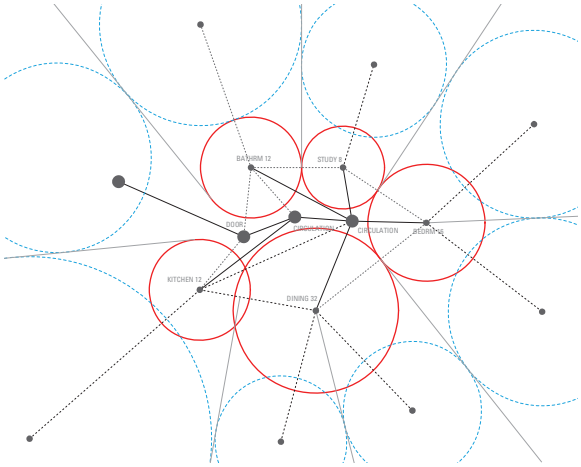
02 PROGRAM-AREA BUBBLES



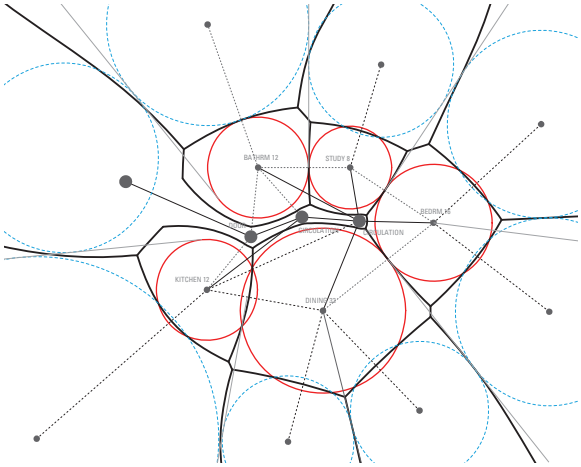
03 REMOVE FRAME STRUCTURE



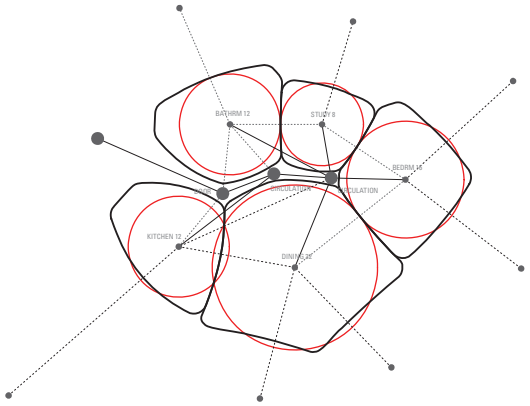
04 REFIT / CIRCLE PACK



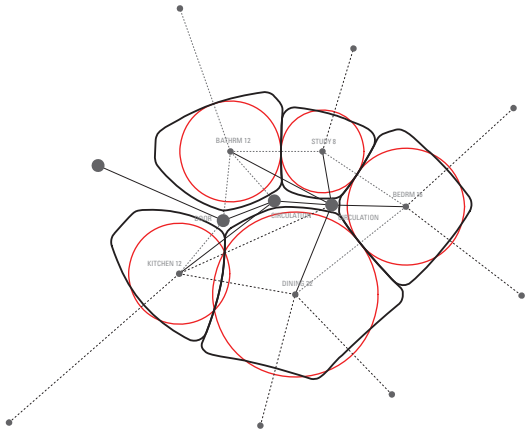
05 OUTBOUND CIRCLES



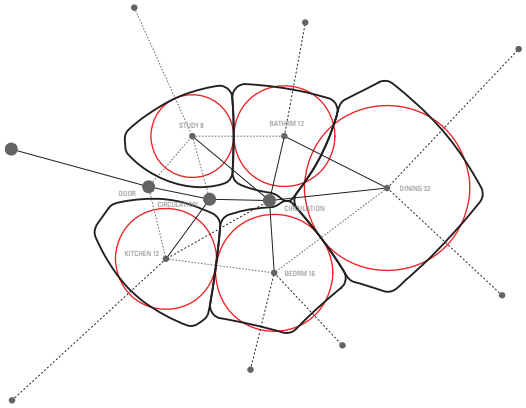
06 CIRCLESET VORONOI



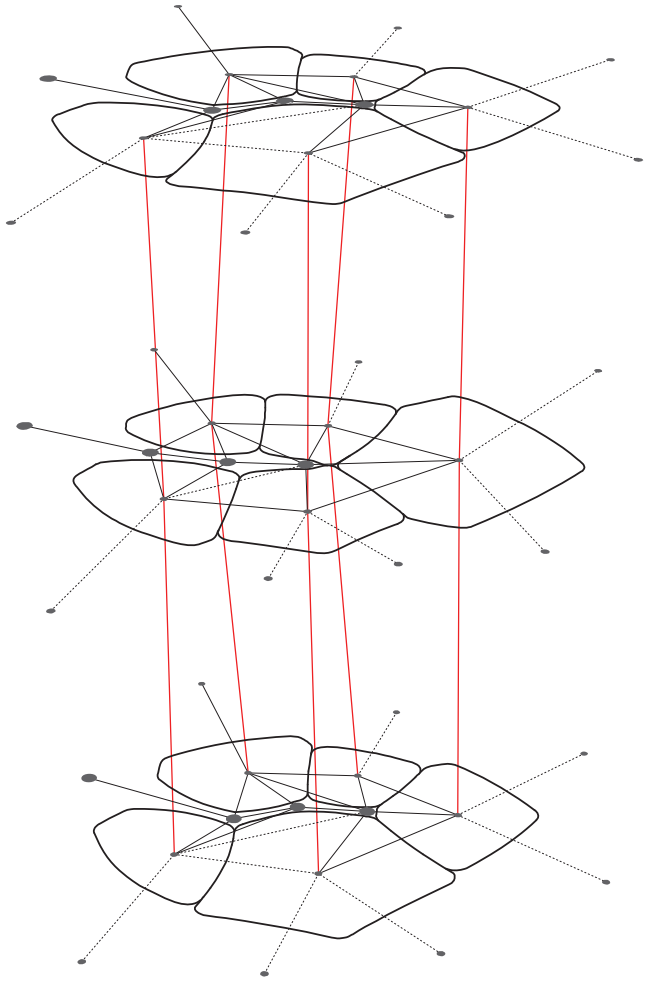
07 PROTOTYPE PLAN



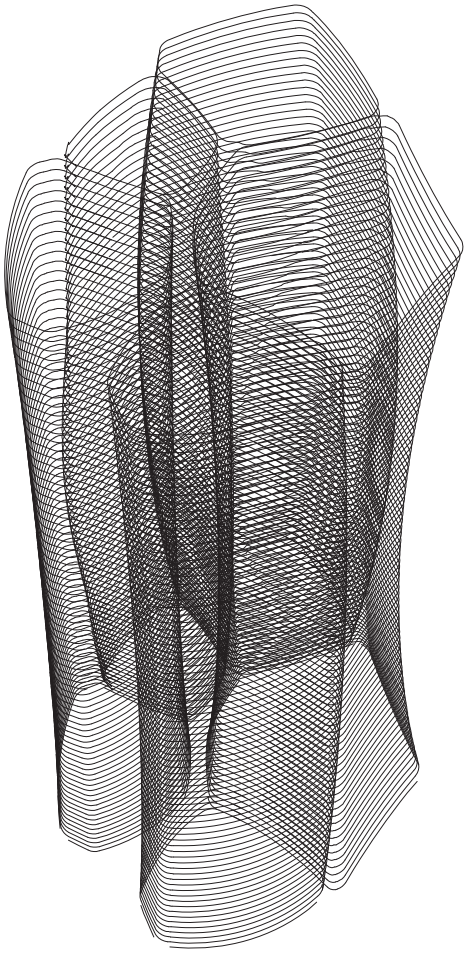
08 PLAN TYPE A - 16 POINT-GRID



09 PLAN TYPE B - 16 POINT-GRID

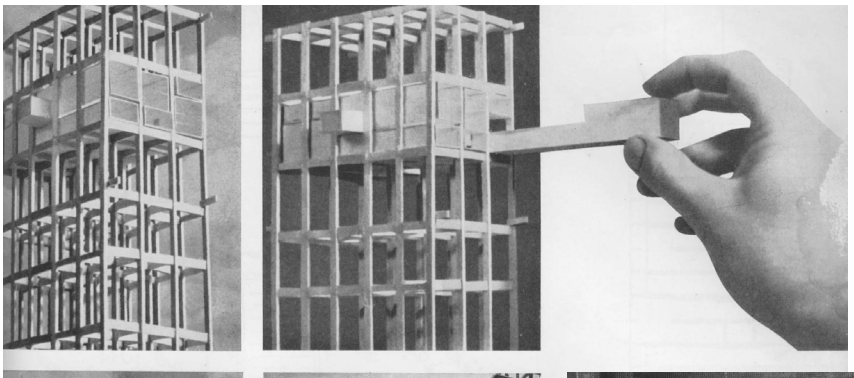


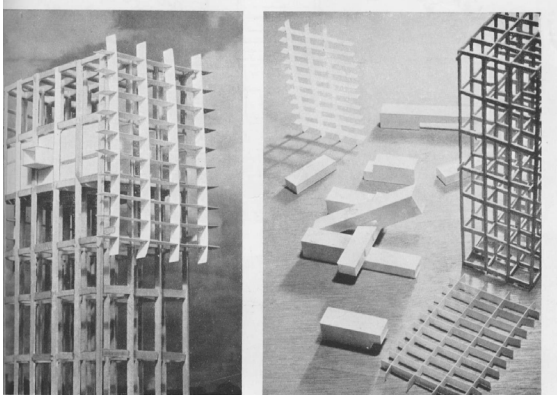
09 3D POINT GRID

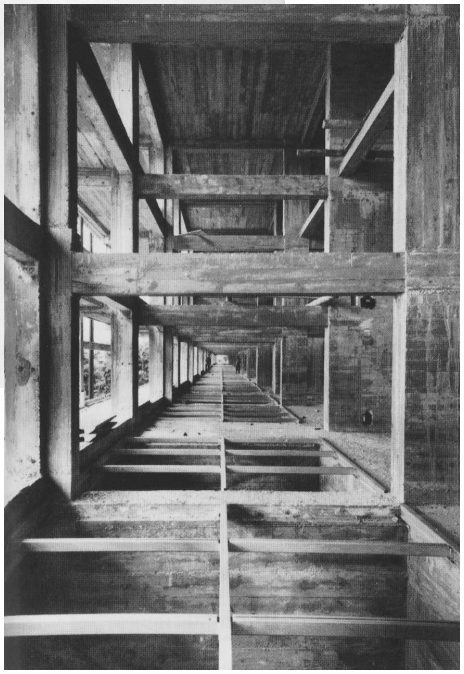


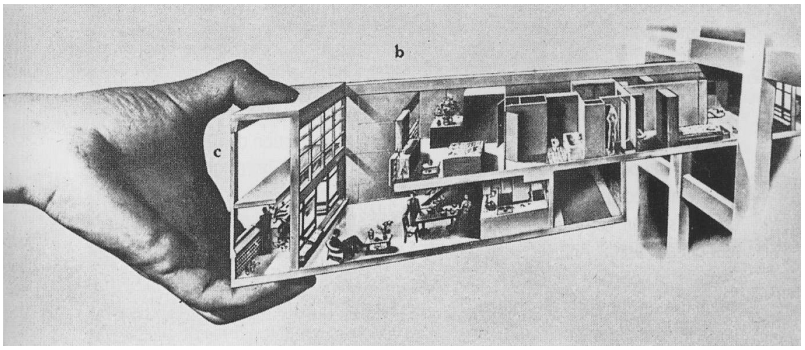
10 LOFT

REORGANIZING THE UNITE D'HABITATION








b
c

In the previous chapter, we investigated the typical building design process and, by getting rid of the process' rigid grid system, developed an alternative prototype consisting of the point grid and program bubbles, including some dummies. In this chapter, we will choose an existing housing project and reorganize it through the prototyping process.

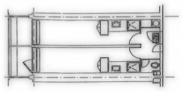
For this purpose, I choose Le Corbusier's Unite d'Hhabitation, an iconic modern housing project heavily reliant on the Domino system. As I mentioned earlier in the previous chapters, Le Coubusier's intention in proposing the Domino system was to use it as a flexible framework that would give architects freedom in their design processes[01]. Although the Unite d'Habitation looks like an aggregation of Domino systems in appearance, it has long been considered a beloved modern design triumph, especially when we look at the beauty of its various unit types and the design freedom within them. It is widely known and acknowledged that this freedom was made possible by the support of Domino system.

At least, that is what we believe.

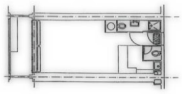
01 Le Corbusier, The Unite d'Habitation in Marseilles by Foundation Le Corbusier
Image Left < Le Corbusier, L'Unite D'Habitation De Marseille by Jacques Sbriglio

Unite Types : Le Corbusier-La cite radieuse de Marseille by Richard A Bisch

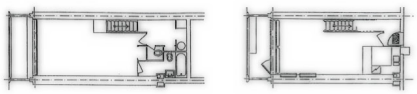
PLAN TYPE A



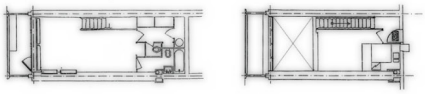
PLAN TYPE B



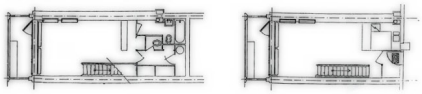
PLAN TYPE CS



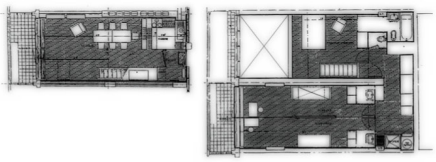
PLAN TYPE CI



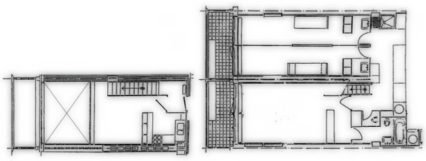
PLAN TYPE CSPI



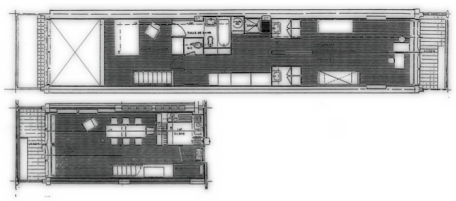
PLAN TYPE E1S



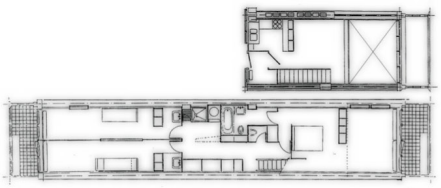
PLAN TYPE E1I



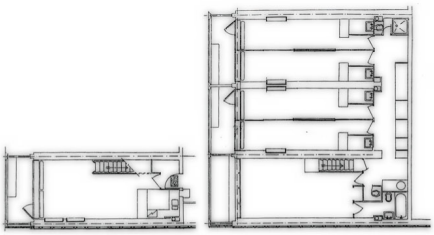
PLAN TYPE E2S



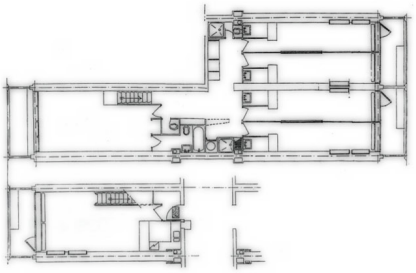
PLAN TYPE E2I



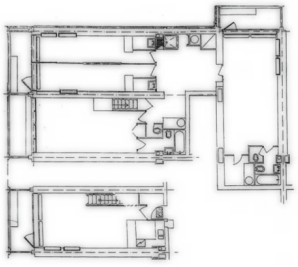
PLAN TYPE G1S



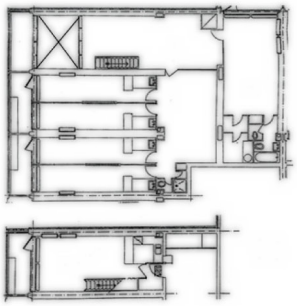
PLAN TYPE G2S



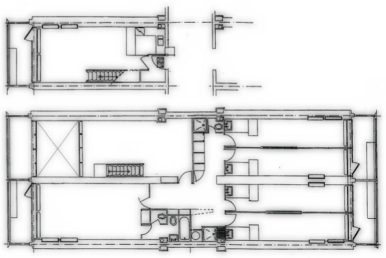
PLAN TYPE G3S



PLAN TYPE H1S

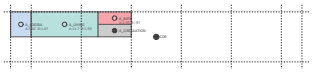


PLAN TYPE H2S

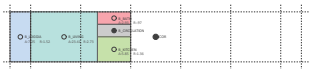


However, in contrast to commonly accepted belief, the Unite d'Habitation was not a product of the harmonious combination of the Domino system as a framework, and various types of units as independent objects sitting freely on top of this framework. Rather, the relationship between these elements is pretty hierarchical, with the Domino system deciding everything from unit types to building shape.

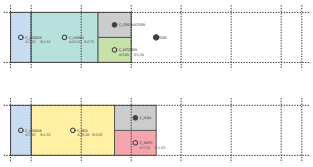
TYPE A



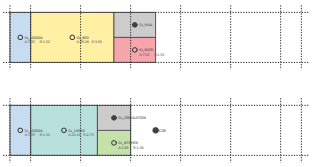
TYPE B



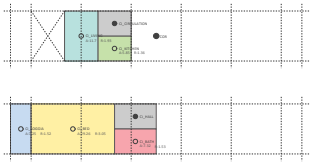
TYPE C



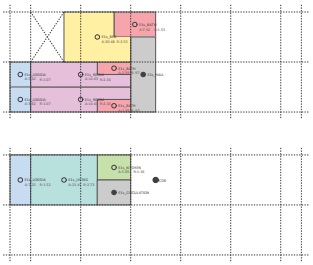
TYPE Cs



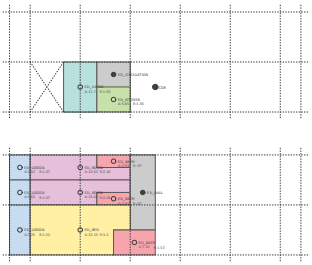
TYPE Ci



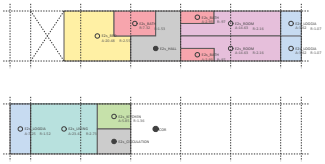
TYPE E1s



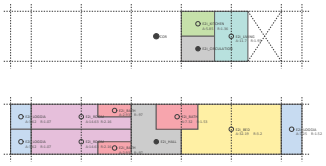
TYPE E1i



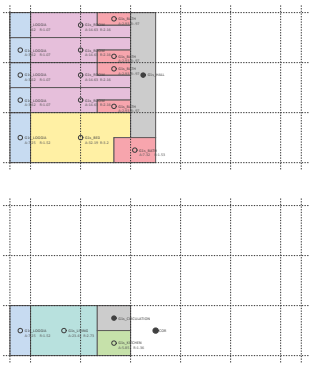
TYPE E2s



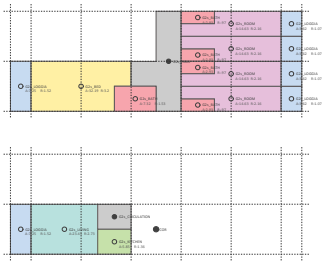
TYPE E2i



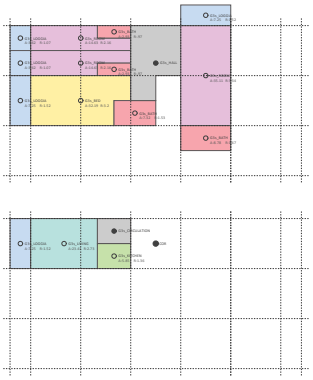
TYPE G1s



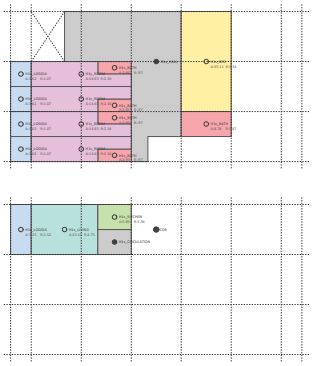
TYPE G2s



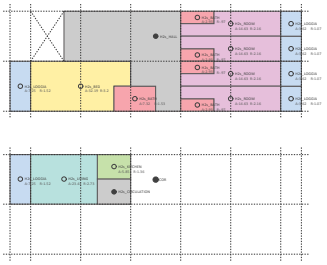
TYPE G3s



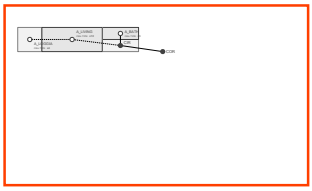
TYPE H1s



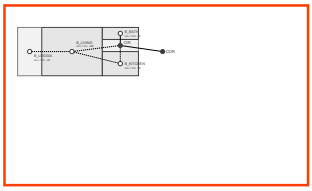
TYPE H2s



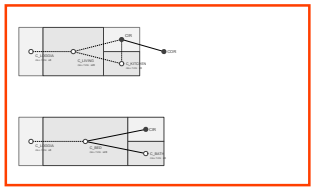
TYPE A



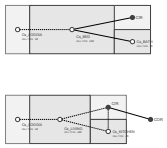
TYPE B



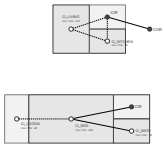
TYPE C



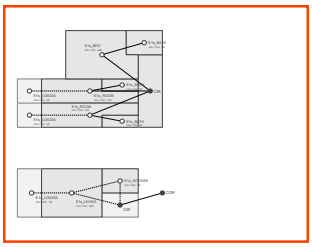
TYPE Cs



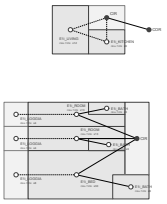
TYPE Ci



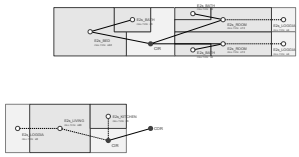
TYPE E1s



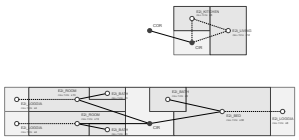
TYPE E1i



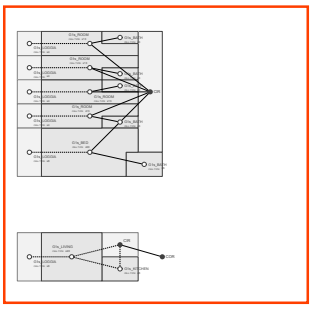
TYPE E2s



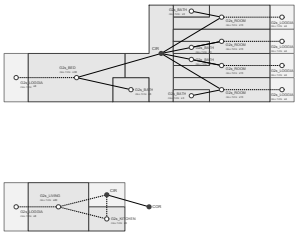
TYPE E2i



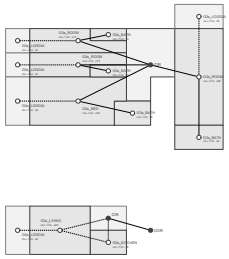
TYPE G1s



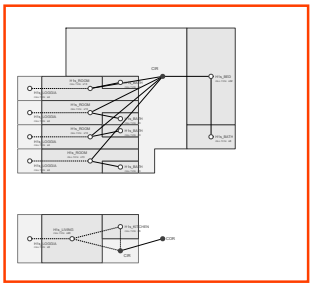
TYPE G2s



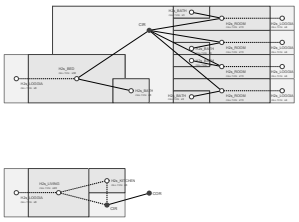
TYPE G3s



TYPE H1s



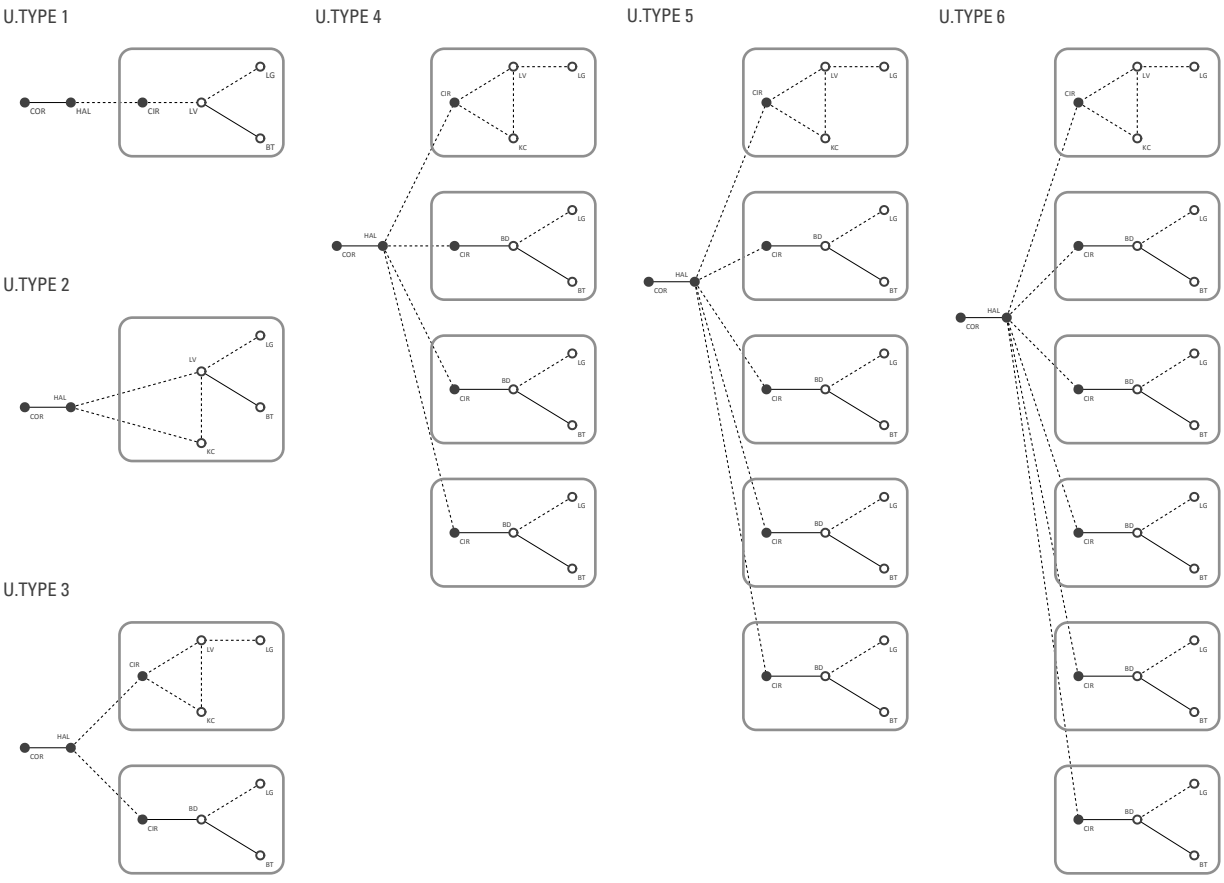
TYPE H2s



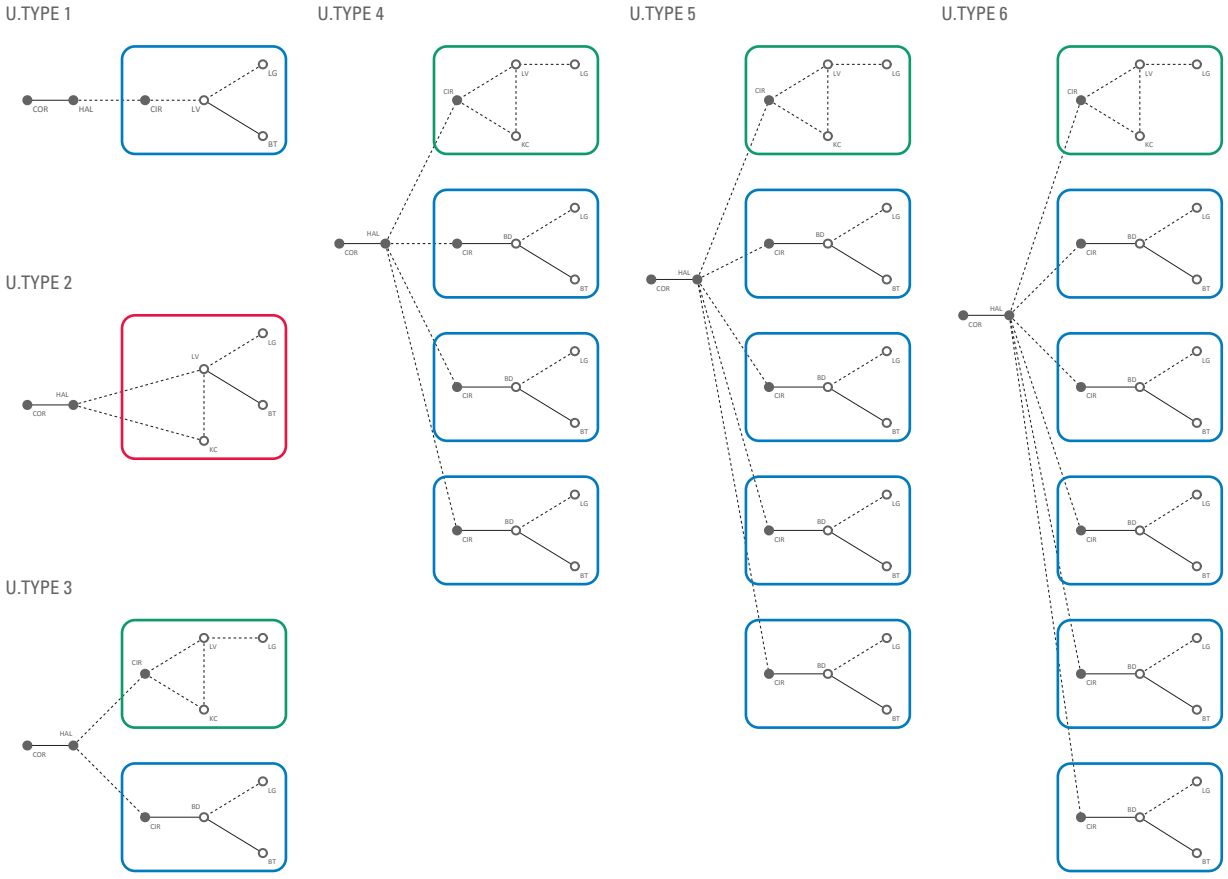
For example, it is believed that in the Unite d'Habitation there are fourteen distinct unit types that are purely unique in terms of their typology [01]. However, the typology diagram shows that there are overlapping/redundant unit types among these fourteen, and when we remove the redundancy, they boil down to only six unique types. This indicates that the fourteen unit types are not the product of typological thoughts or needs, but of the ad-hoc, situational needs arising from the need to morph them into the grid system.

In the simplified network diagram below, we can easily see that there are only six unique unit types.

01 Le Corbusier-La cite radieuse de Marseille by Richard A Bisch

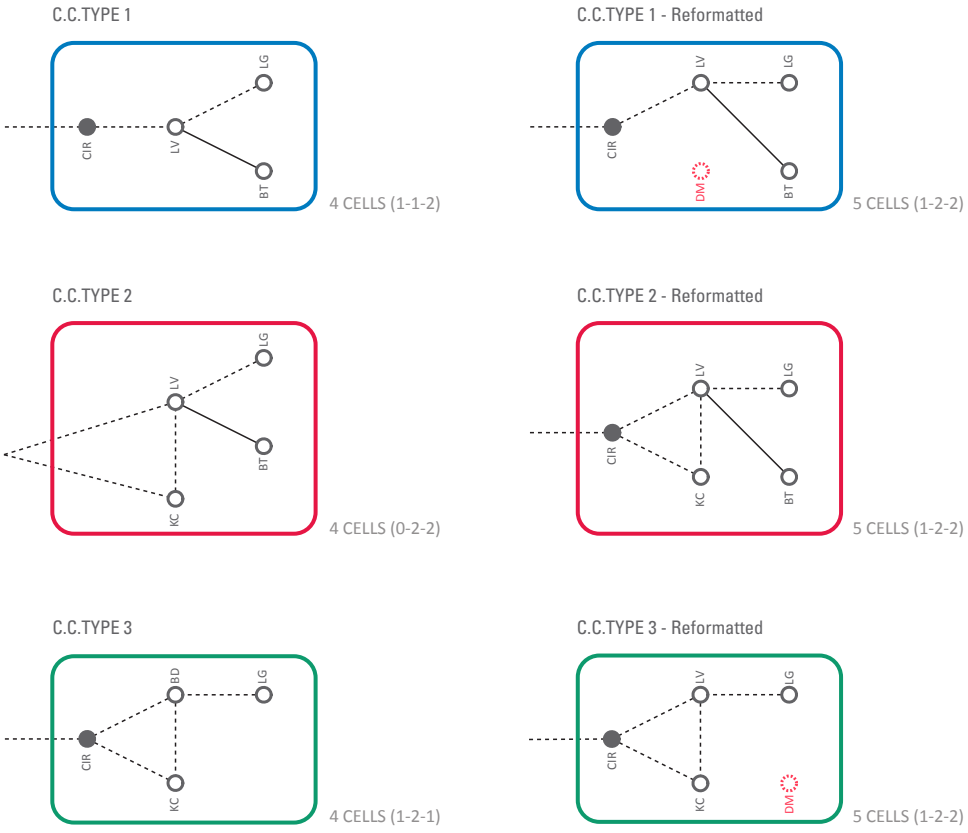


In addition, these six types share some parts, meaning that they are the product of mixing and matching three unique cell cluster types. As we can see from the Diagram 1, they are composed of four cells. However, their formations are slightly different from one another. For example, Cell Cluster Type 1 (C.C. TYPE1) has one cell in the first place and another cell in the next, followed by two other cells, creating a 1-1-2 format. Cell Cluster Type 2 (C.C. TYPE 2) has a 0-2-2 format, while C.C. TYPE 3 has a 1-2-1.



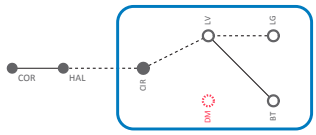
As is discussed in the prototype chapter, the different number of cells in the clusters will cause a significant problem in lofting and bundling them together. That being said, we somehow need to match the number of cells and their formations, which can be easily done by adding some dummy cells to the clusters.

Adding dummy cells will make the clusters have the same numbers of cells and exactly the same cell formation, in this case, 1-2-2. This formation will be the smallest scale, non-divisible framework on top of which program bubbles can be mapped.

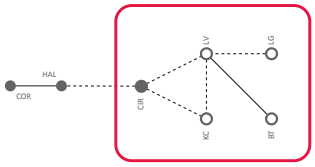


The diagram shows the entire map of the six unit types and how these are organized by the three cell-cluster types.

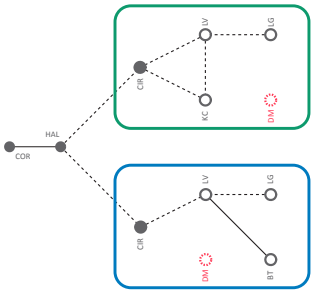
U.TYPE 1
COR.HAL.01(CIR.LV.KC.LG.DM)



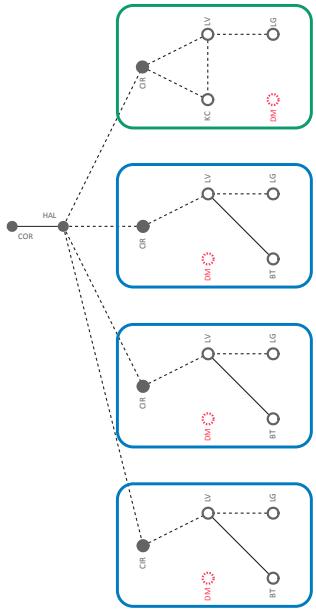
U.TYPE 2
COR.HAL.01(CIR.LV.KC.LG.BT)



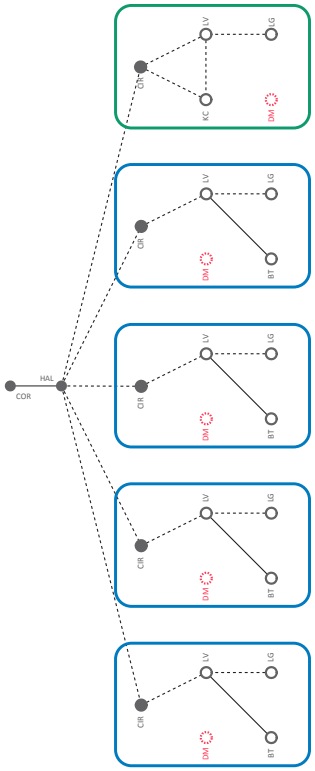
U.TYPE 3
COR.HAL.01(CIR.LV.KC.LG.DM).01(CIR.LV.DM.LG.BT)



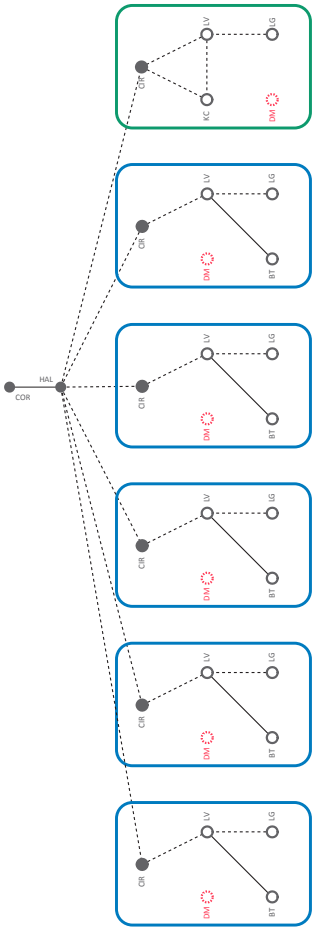
U.TYPE 4
COR.HAL.01(CIR.LV.KC.LG.DM).03(CIR.LV.DM.LG.BT)



U.TYPE 5
COR.HAL.01(CIR.LV.KC.LG.DM).04(CIR.LV.DM.LG.BT)



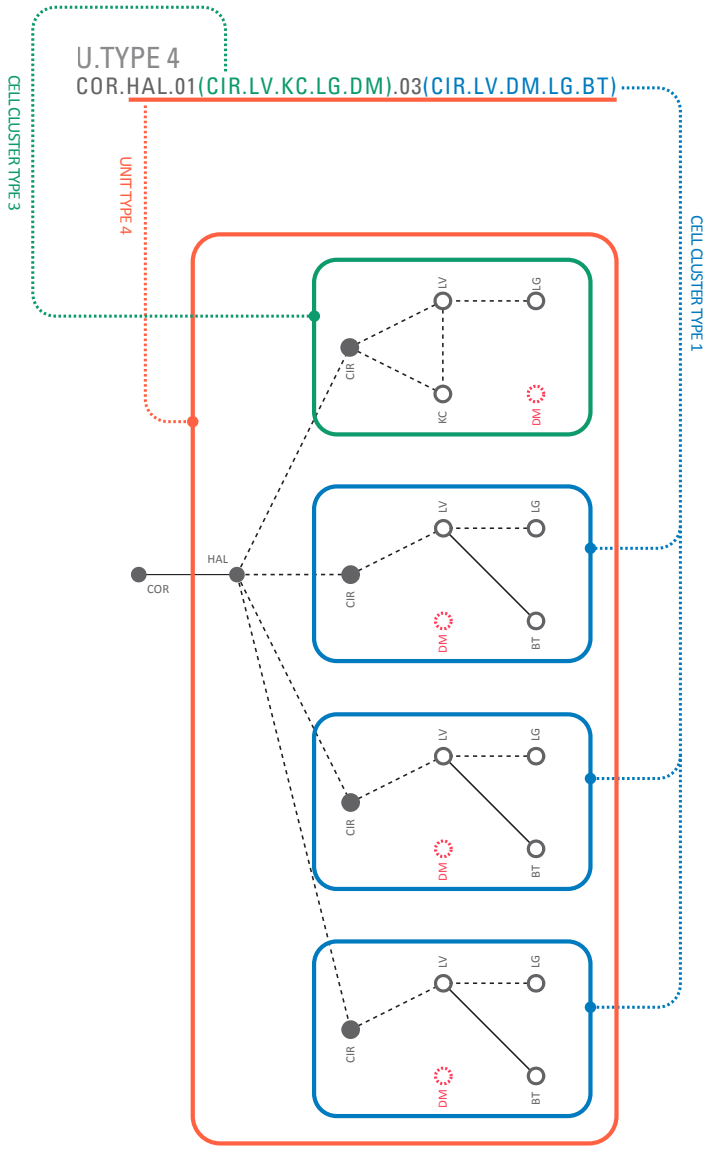
U.TYPE 6
COR.HAL.01(CIR.LV.KC.LG.DM).05(CIR.LV.DM.LG.BT)



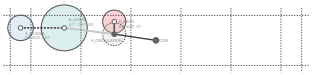
For example, a three-bedroom apartment type can be expressed with COR.HAL.01(CIR.LV.KC.LG.DM).03(CIR.LV.DM.LG.BT),

where (CIR.LV.KC.LG.DM) represents a cell cluster of one circulation, living room, kitchen, and loggia; and

03(CIR.LV.DM.LG.BT) represents three cell clusters each consisting of one circulation, bedroom, bathroom, and loggia.

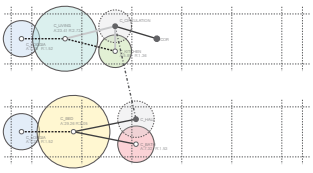


TYPE A

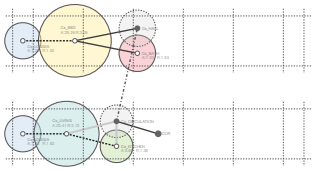


TYPE B

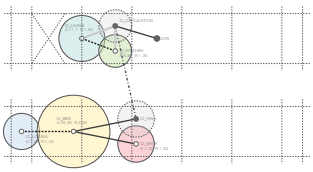
TYPE C



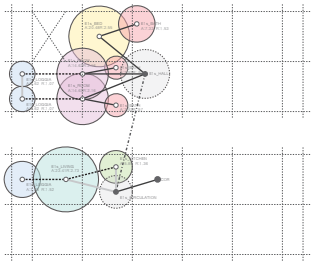
TYPE Cs



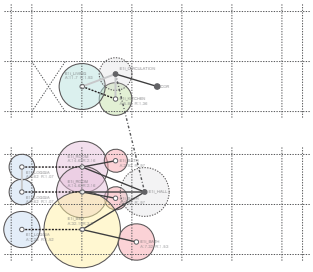
TYPE Ci



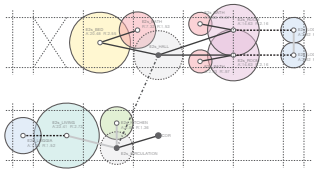
TYPE E1s



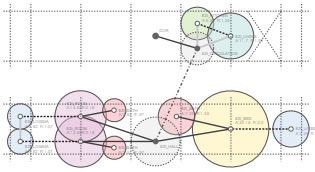
TYPE E1i



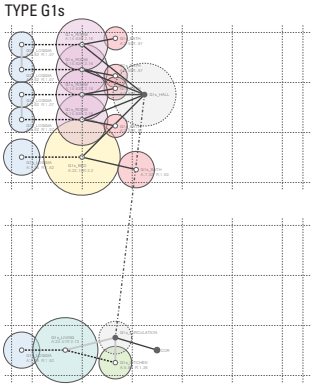
TYPE E2s



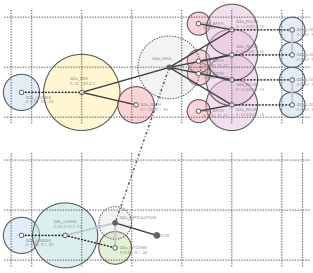
TYPE E2i



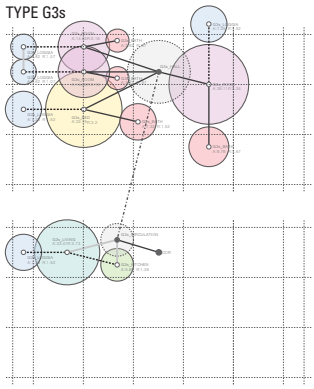
TYPE G1s



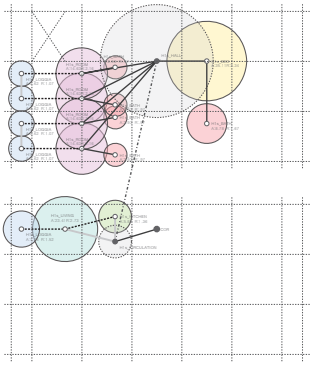
TYPE G2s



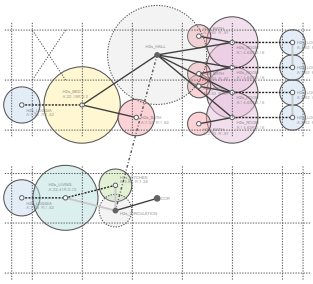
TYPE G3s



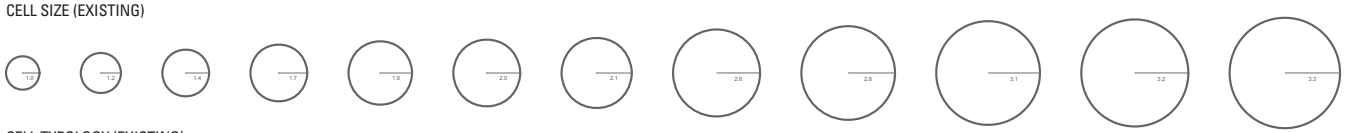
TYPE H1s



TYPE H2s



CELL SIZE (EXISTING)



CELL TYPOLOGY (EXISTING)

CIR HAL COR LV LG BT KC BD

CELL SIZE (PROPOSED)

CIR

HAL / COR

DUMMY CELL
FLEXIABLE RADIUS

KT / BT

SMALL CELL
RADIUS 1.5 OR BELOW

LG

MEDIUM CELL
RADIUS 1.6 TO 2.0

LV / BD / RM

LARGE CELL
RADIUS 2.5

In the previous discussion, we established the framework for the alternative design process and also mapped required programs on the framework. Now we need to figure out the size of individual program bubbles based on their actual footprints.

From the room data sheet, we can get twelve different sizes of program bubbles that fall into roughly five categories; points without area data, dummy bubbles with a flexible area, small (radius of 1.5 meters or below), medium (radius from 1.6 – 2.0 meters) and large (radius from 2.1 – 2.5 meters) bubbles, depending on the programs they contain.

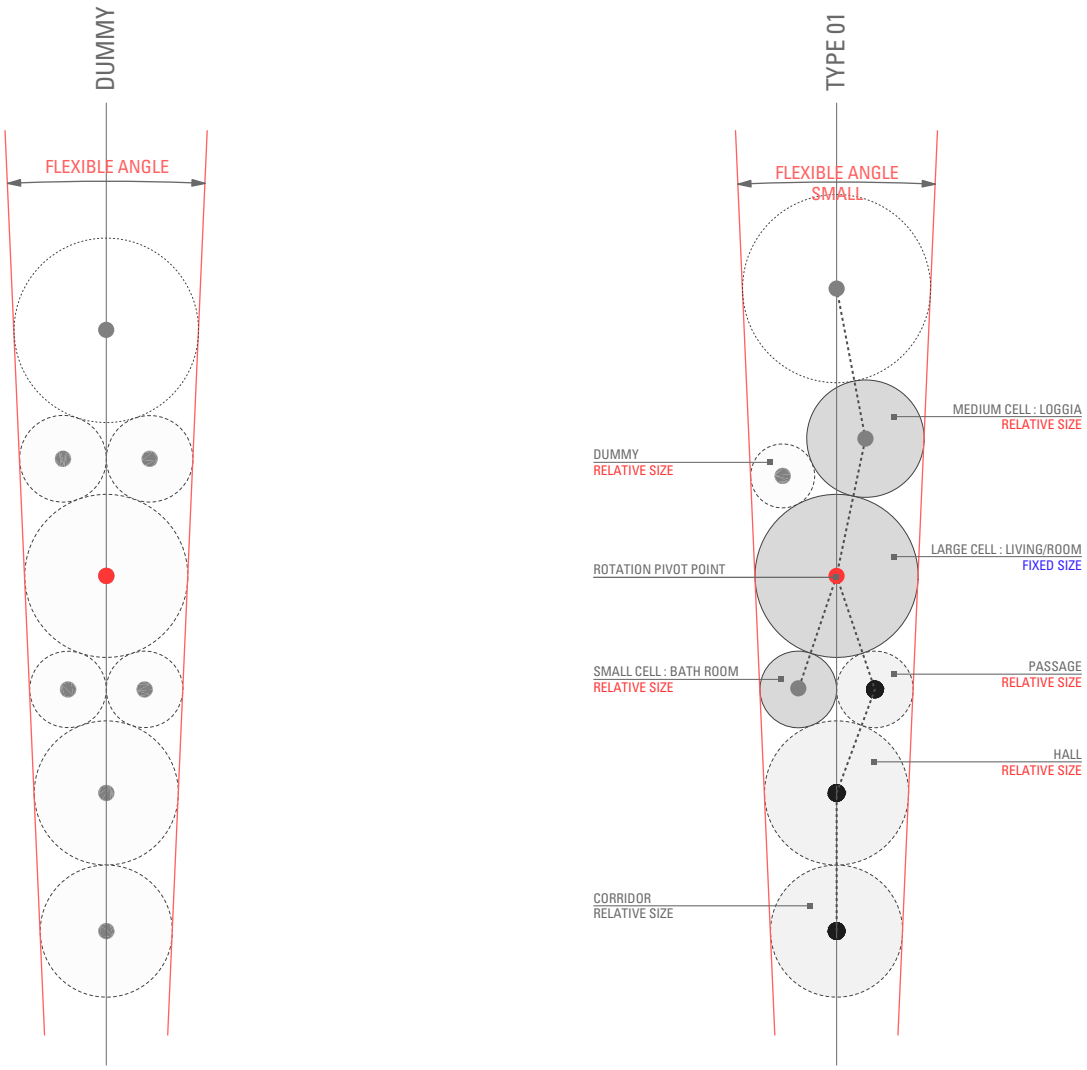
A point without area data, represented by single dot, is for circulation that usually does not have any areal requirement. For planning efficiency, circulation areas usually need to be minimized. Dummy bubbles can represent a hallway or a closet space, depending on how the overall plan looks. Small-sized bubbles with a radius of less than 1.5 meters can accommodate a kitchen or bathroom. Medium-sizes bubble with radius between 1.6 and 2.0 meters can be the best spot for a loggia, and large-sized ones with a radius bigger than 2.1 meters, for a bedroom or living room.

UNIT TYPE	CELL NAME	AREA	RADIUS	CELL TYPES x03	x04	x06	x08	x10	x12	x13	x20	x22	x28	x30	x32	UNIT CODE (BY PROGRAM)	EA
A	A_BATH	3.30	1.00													01(CIR.LV.BT.LG)	8
	A_LOGGIA	4.40	1.20														
	A_LIVING	11.00	1.90														
	A_CIRCULATION																
	A_HALL																
B	B_BATH	3.30	1.00													01(LV.KT.BT.LG)	26
	B_KITCHEN	6.60	1.40														
	B_LOGGIA	8.80	1.70														
	B_LIVING	22.00	2.60														
	B_CIRCULATION																
	B_HALL																
C	C_BATH	6.60	1.40													01(CIR.LV.KC.LG).01(CIR.BD.BT.LG)	7
	C_KITCHEN	6.60	1.40														
	C_LOGGIA	8.80	1.70														
	C_LOGGIA	8.80	1.70														
	C_LIVING	22.00	2.60														
	C_BED	30.80	3.10														
	C_CIRCULATION																
	C_HALL																
Ci	Ci_BATH	6.60	1.40													01(CIR.LV.KC.LG).01(CIR.BD.BT.LG)	45
	Ci_KITCHEN	6.60	1.40														
	Ci_LOGGIA	8.80	1.70														
	Ci_LIVING	13.20	2.00														
	Ci_BED	30.80	3.10														
	Ci_CIRCULATION																
	Ci_HALL																
Cs	Cs_BATH	6.60	1.40													01(CIR.LV.KC.LG).01(CIR.BD.BT.LG)	6
	Cs_KITCHEN	6.60	1.40														
	Cs_LOGGIA	8.80	1.70														
	Cs_LOGGIA	8.80	1.70														
	Cs_LIVING	22.00	2.60														
	Cs_BED	30.80	3.10														
	Cs_CIRCULATION																
	Cs_HALL																
E1i	E1i_BATH	3.30	1.00													01(CIR.LV.KC.LG).03(CIR.BD.BT.LG)	13
	E1i_BATH	3.30	1.00														
	E1i_LOGGIA	4.40	1.20														
	E1i_LOGGIA	4.40	1.20														
	E1i_BATH	6.60	1.40														
	E1i_KITCHEN	6.60	1.40														
	E1i_LOGGIA	8.80	1.70														
	E1i_LIVING	13.20	2.00														
	E1i_ROOM	14.30	2.10														
	E1i_ROOM	14.30	2.10														
	E1i_BED	33.00	3.20														
	E1i_CIRCULATION																
	E1i_HALL																
E1s	E1s_BATH	3.30	1.00													01(CIR.LV.KC.LG).03(CIR.BD.BT.LG)	7
	E1s_BATH	3.30	1.00														
	E1s_LOGGIA	4.40	1.20														
	E1s_LOGGIA	4.40	1.20														
	E1s_BATH	6.60	1.40														
	E1s_KITCHEN	6.60	1.40														
	E1s_LOGGIA	8.80	1.70														
	E1s_ROOM	14.30	2.10														
	E1s_ROOM	14.30	2.10														
	E1s_LIVING	22.00	2.60														
	E1s_BED	24.20	2.80														
	E1s_CIRCULATION																
	E1s_HALL																
E2i	E2i_BATH	3.30	1.00													01(CIR.LV.KC.LG).03(CIR.BD.BT.LG)	111
	E2i_BATH	3.30	1.00														
	E2i_LOGGIA	4.40	1.20														
	E2i_LOGGIA	4.40	1.20														
	E2i_BATH	6.60	1.40														
	E2i_KITCHEN	6.60	1.40														
	E2i_LOGGIA	8.80	1.70														
	E2i_LIVING	13.20	2.00														
	E2i_ROOM	14.30	2.10														
	E2i_ROOM	14.30	2.10														
	E2i_BED	33.00	3.20														
	E2i_CIRCULATION																
	E2i_HALL																
E2s	E2s_BATH	3.30	1.00													01(CIR.LV.KC.LG).03(CIR.BD.BT.LG)	89
	E2s_BATH	3.30	1.00														
	E2s_LOGGIA	4.40	1.20														
	E2s_LOGGIA	4.40	1.20														
	E2s_BATH	6.60	1.40														
	E2s_KITCHEN	6.60	1.40														
	E2s_LOGGIA	8.80	1.70														
	E2s_ROOM	14.30	2.10														
	E2s_LIVING	22.00	2.60														
	E2s_BED	24.20	2.80														
	E2s_CIRCULATION																
	E2s_HALL																

UNIT TYPE	CELL NAME	AREA	RADIUS	CELL TYPES x03	x04	x06	x08	x10	x12	x13	x20	x22	x28	x30	x32	UNIT CODE (BY PROGRAM)	EA
G1s	G1s_BATH	3.30	1.00													01(CIR.LV.KC.LG).05(CIR.BD.BT.LG)	3
	G1s_BATH	3.30	1.00														
	G1s_BATH	3.30	1.00														
	G1s_BATH	3.30	1.00														
	G1s_LOGGIA	4.40	1.20														
	G1s_LOGGIA	4.40	1.20														
	G1s_LOGGIA	4.40	1.20														
	G1s_LOGGIA	4.40	1.20														
	G1s_BATH	6.60	1.40														
	G1s_KITCHEN	6.60	1.40														
	G1s_LOGGIA	8.80	1.70														
	G1s_LOGGIA	8.80	1.70														
	G1s_ROOM	14.30	2.10														
	G1s_ROOM	14.30	2.10														
	G1s_ROOM	14.30	2.10														
	G1s_LIVING	22.00	2.60														
	G1s_BED	33.00	3.20														
	G1s_CIRCULATION																
	G1s_HALL																
G2s	G2s_BATH	3.30	1.00													01(CIR.LV.KC.LG).05(CIR.BD.BT.LG)	15
	G2s_BATH	3.30	1.00														
	G2s_BATH	3.30	1.00														
	G2s_BATH	3.30	1.00														
	G2s_LOGGIA	4.40	1.20														
	G2s_LOGGIA	4.40	1.20														
	G2s_LOGGIA	4.40	1.20														
	G2s_LOGGIA	4.40	1.20														
	G2s_BATH	6.60	1.40														
	G2s_KITCHEN	6.60	1.40														
	G2s_LOGGIA	8.80	1.70														
	G2s_LOGGIA	8.80	1.70														
	G2s_ROOM	14.30	2.10														
	G2s_ROOM	14.30	2.10														
	G2s_ROOM	14.30	2.10														
	G2s_ROOM	14.30	2.10														
	G2s_LIVING	22.00	2.60														
	G2s_BED	33.00	3.20														
	G2s_CIRCULATION																
	G2s_HALL																
G3s	G3s_BATH	3.30	1.00													01(CIR.LV.KC.LG).04(CIR.BD.BT.LG)	2
	G3s_BATH	3.30	1.00														
	G3s_LOGGIA	4.40	1.20														
	G3s_LOGGIA	4.40	1.20														
	G3s_BATH	6.60	1.40														
	G3s_KITCHEN	6.60	1.40														
	G3s_BATH	8.80	1.70														
	G3s_LOGGIA	8.80	1.70														
	G3s_LOGGIA	8.80	1.70														
	G3s_LOGGIA	8.80	1.70														
	G3s_ROOM	14.30	2.10														
	G3s_ROOM	14.30	2.10														
	G3s_LIVING	22.00	2.60														
	G3s_BED	33.00	3.20														
	G3s_ROOM	35.20	3.30														
	G3s_CIRCULATION																
	G3s_HALL																
H1s	H1s_BATH	3.30	1.00													01(CIR.LV.KC.LG).05(CIR.BD.BT.LG)	1
	H1s_BATH	3.30	1.00														
	H1s_BATH	3.30	1.00														
	H1s_BATH	3.30	1.00														
	H1s_LOGGIA	4.40	1.20														
	H1s_LOGGIA	4.40	1.20														
	H1s_LOGGIA	4.40	1.20														
	H1s_LOGGIA	4.40	1.20														
	H1s_KITCHEN	6.60	1.40														
	H1s_BATH	8.80	1.70														
	H1s_LOGGIA	8.80	1.70														
	H1s_ROOM	14.30	2.10														
	H1s_ROOM	14.30	2.10														
	H1s_ROOM	14.30	2.10														
	H1s_ROOM	14.30	2.10														
	H1s_LIVING	22.00	2.60														
	H1s_BED	35.20	3.30														
	H1s_CIRCULATION																
	H1s_HALL																
H2s	H2s_BATH	3.30	1.00													01(CIR.LV.KC.LG).05(CIR.BD.BT.LG)	1
	H2s_BATH	3.30	1.00														
	H2s_BATH	3.30	1.00														
	H2s_BATH	3.30	1.00														
	H2s_LOGGIA	4.40	1.20														
	H2s_LOGGIA	4.40	1.20														
	H2s_LOGGIA	4.40	1.20														
	H2s_LOGGIA	4.40	1.20														
	H2s_BATH	6.60	1.40														
	H2s_KITCHEN	6.60	1.40														
	H2s_LOGGIA	8.80	1.70														
	H2s_LOGGIA	8.80	1.70														
	H2s_ROOM	14.30	2.10														
	H2s_ROOM	14.30	2.10														
	H2s_ROOM	14.30	2.10														
	H2s_ROOM	14.30	2.10														
	H2s_LIVING	22.00	2.60														
	H2s_BED	33.00	3.20														
	H2s_CIRCULATION																
	H2s_HALL																
				28	27	24	22	1	3	26	10	2	3	6	2		

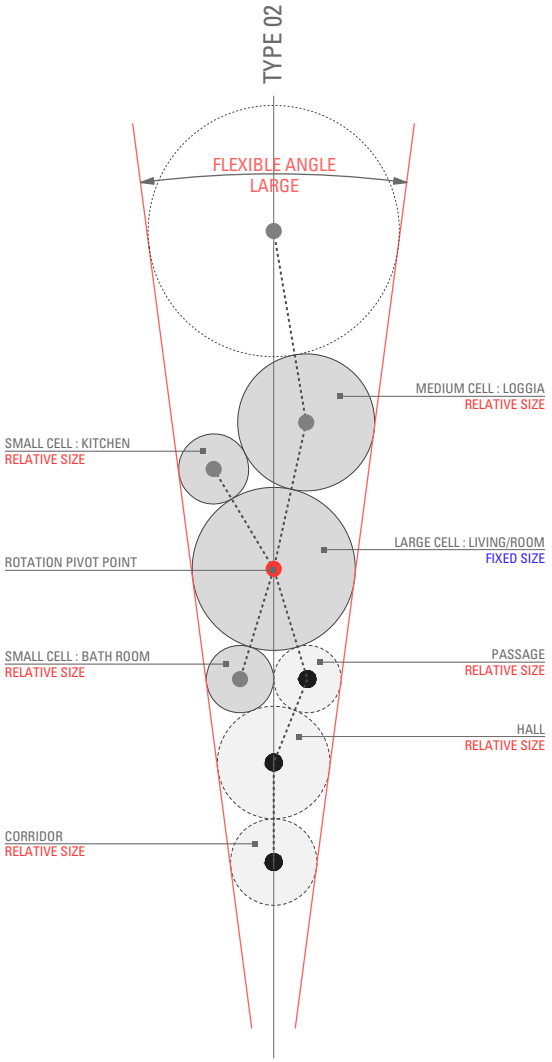
By combining bubble-size types and cell-cluster types (pinpoint connection types), we can get the four complete cell cluster types. Each type is flexible because the radius of small, medium, or large bubbles is not decisive, but has a little room for changing, and because the dummy cells are totally flexible in their size as well.

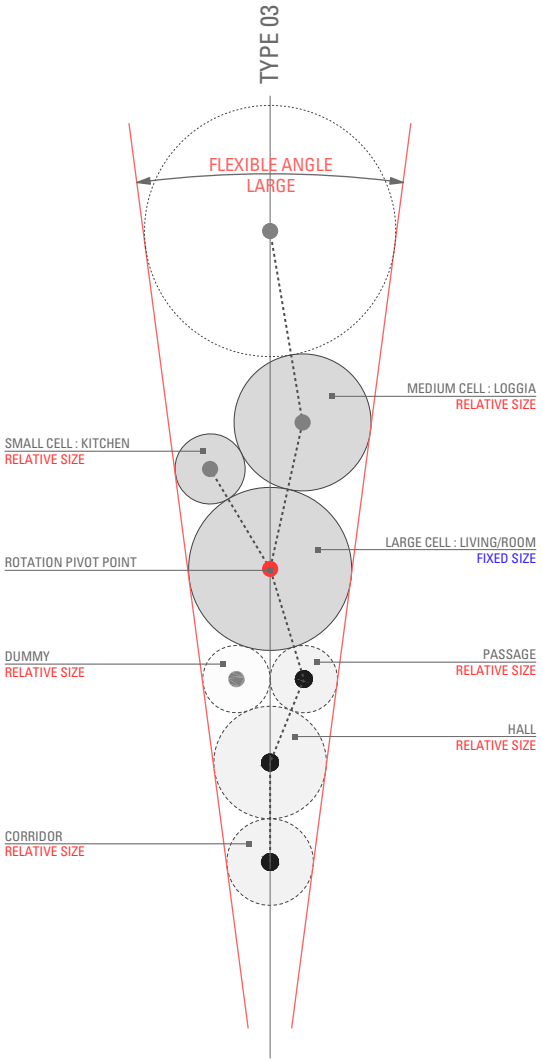
To make them easier to control, large cells for living rooms / bedrooms are moved to the center of cell clusters. Since their range of size fluctuation is relatively less than those of medium or small cells, we set their size as fixed so we can use them as pivotal points in the cell clusters.

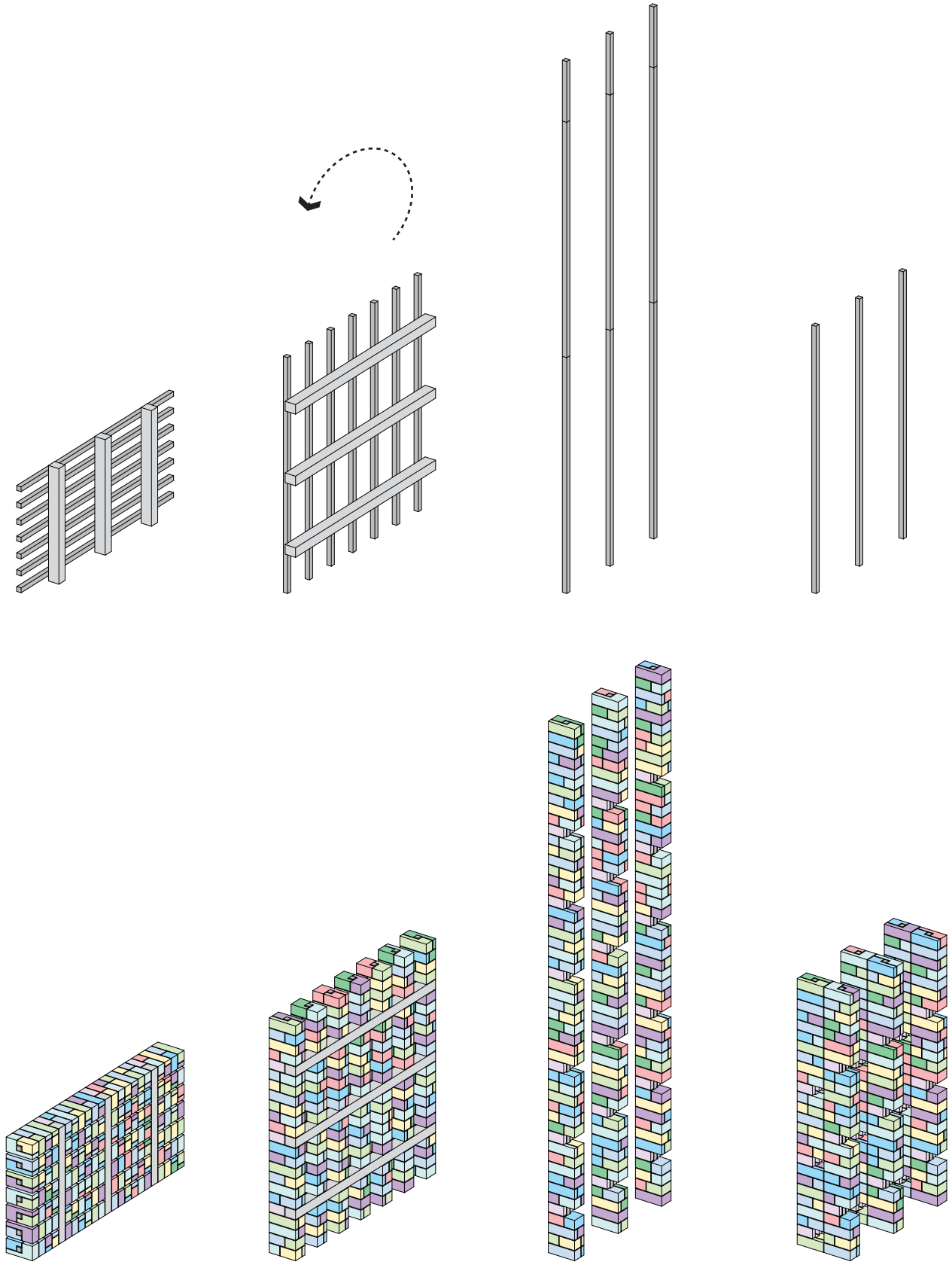


Then based on the way each cell cluster was organized in the previous discussion, we rearrange other cells while maintaining their original relationship with the large cells. We also need to add one more flexible-sized cell at the top of the cell cluster diagram, which will help us to define the outer boundary of building when we apply circle sets with Voronoi logic. In addition, to make shared connection points between cell clusters, we add two more cells at the bottom of the diagram, one for internal circulation and the other for external circulation. Red lines represent the boundary of cell clusters that will be shared with the adjacent cell cluster types.

Now four cell cluster types can be flexible, with one pivotal point at the center marked in red.







EXISTING CONDITION
(3 CORES / 7 CORRIDORS)

ROTATED
(7 CORES / 3 CORRIDORS)

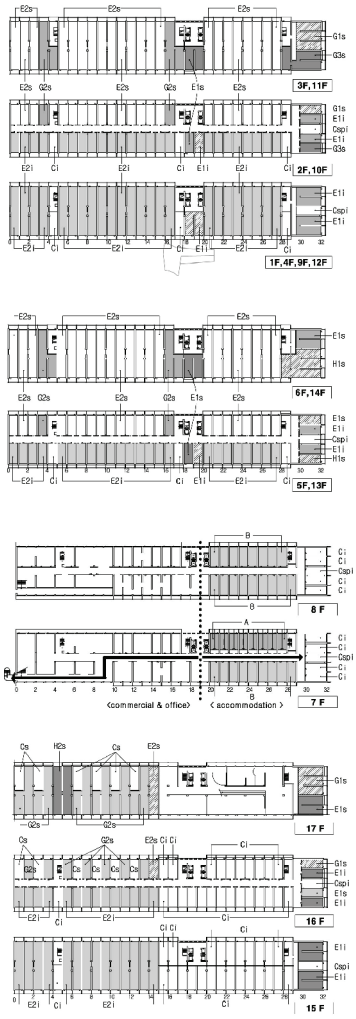
MERGED
(3 CORES)

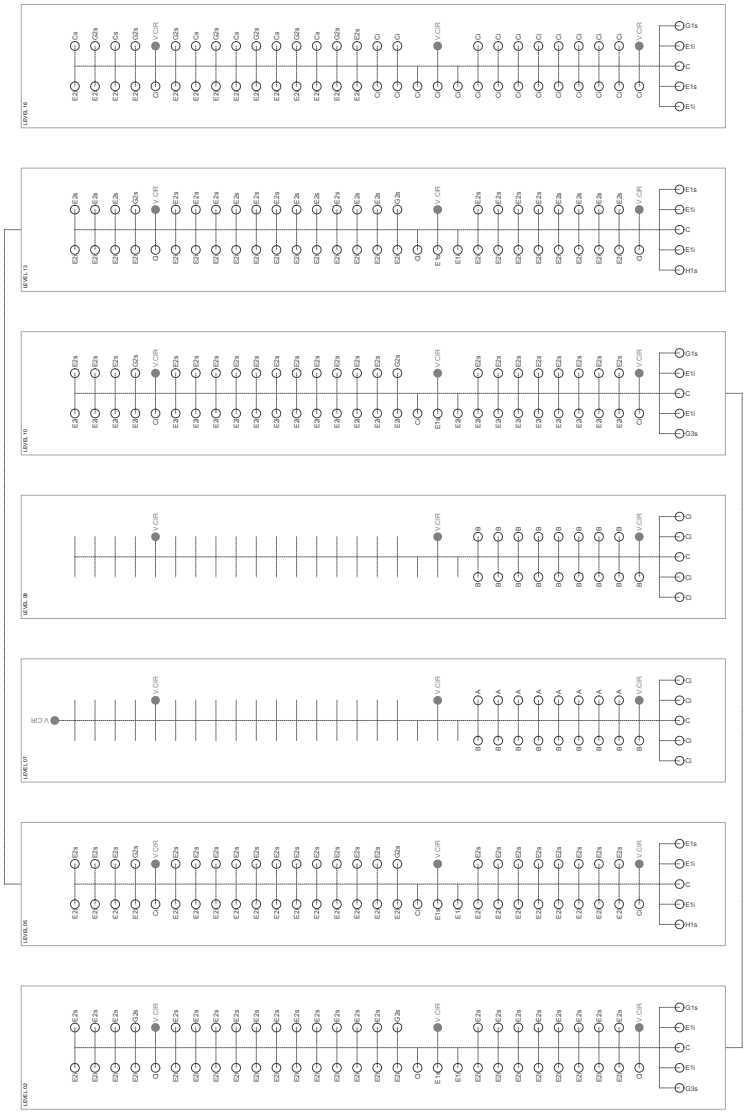
MERGED
(3 CORES)

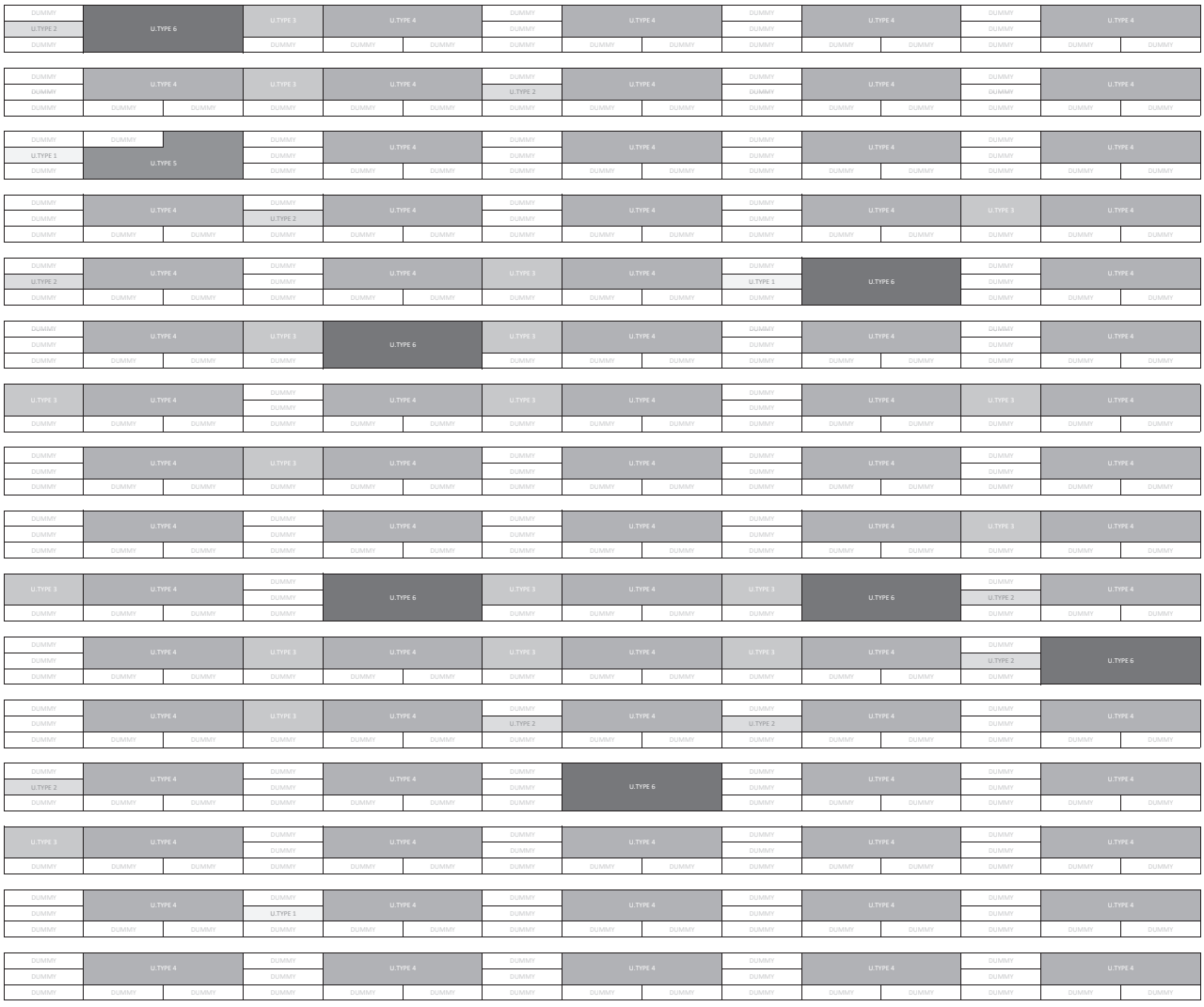
note : diagram doesn't represent building factors such as numbers of unit types

Examining the existing structure of the building in terms of how individual units are tied together will help to reorganize those four cell clusters into a whole building. The first thing that we notice from the diagram of the existing floor plan is that there are three main cores in the building. The second thing might be the fact that only seven floors have corridors, though the building is seventeen stories high, because of how units, including duplexes, are organized vertically. It would be a good starting point to consider keeping three main cores while maintaining the way in which units are attached or bound to the corridors.

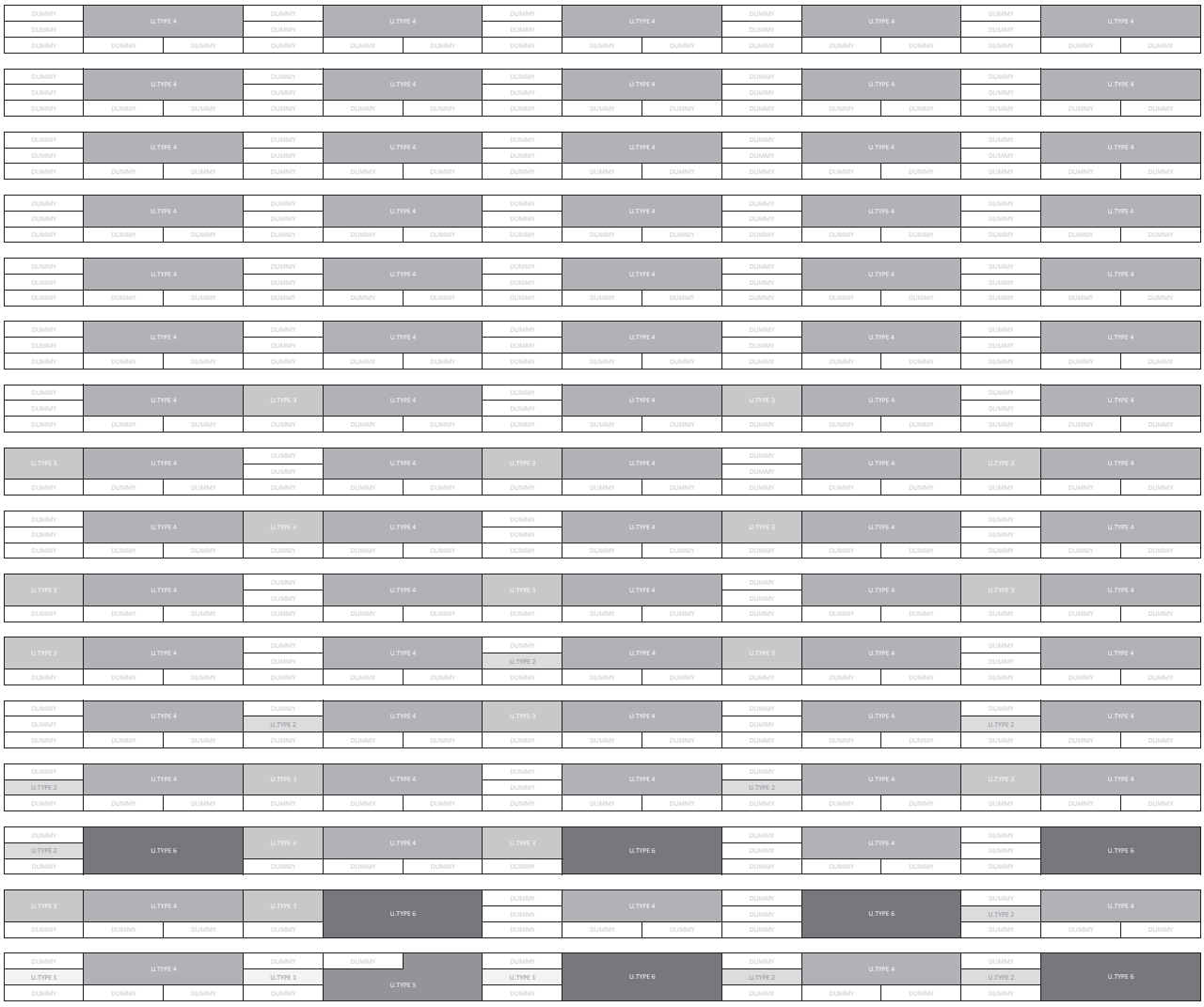
For this, we will first horizontally rotate long corridors by 90 degrees to make them vertical. Then we will combine those seven corridors into three by mixing and matching.






DUMMY
U.TYPE 2
U.TYPE 6
U.TYPE 3
U.TYPE 4
U.TYPE 1
U.TYPE 5
U.TYPE 6

As we can see from the diagram on the left, these vertical circulations are not efficient in terms of flow. If we can relocate small units to the bottom and big ones to the top, the circulation becomes more efficient. The table in the left shows the existing unit array where different units intermingle in random fashion to morph themselves into the physical frame. The one in the right, on the other hand, is a rearranged unit array diagram based on their size, where we can see the optimized vertical circulation flow.



UNIT TYPE 1
3EA / BUNDLE

U.TYPE 1

C.C.TYPE 1 < ENT

UNIT TYPE 2
9 EA / BUNDLE

U.TYPE 2

C.C.TYPE 2 < ENT

UNIT TYPE 3
20 EA / BUNDLE

U.TYPE 3

C.C.TYPE 1
C.C.TYPE 3 < ENT

UNIT TYPE 4
77 EA / BUNDLE

U.TYPE 4

C.C.TYPE 1 | C.C.TYPE 1
C.C.TYPE 1 | C.C.TYPE 3 < ENT

UNIT TYPE 5
1 EA / BUNDLE

U.TYPE 5

C.C.TYPE 1
C.C.TYPE 1 | C.C.TYPE 3 < ENT
C.C.TYPE 1 | C.C.TYPE 1

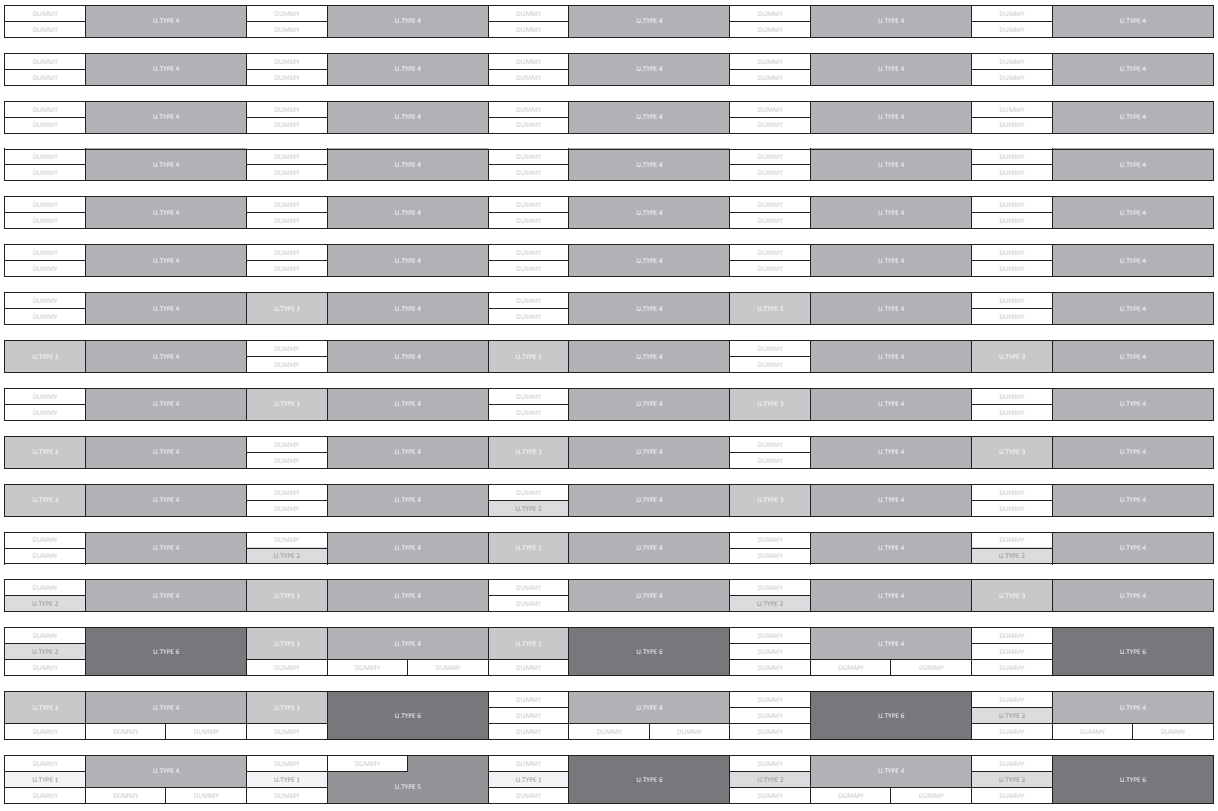
UNIT TYPE 6
7 EA / BUNDLE

U.TYPE 6

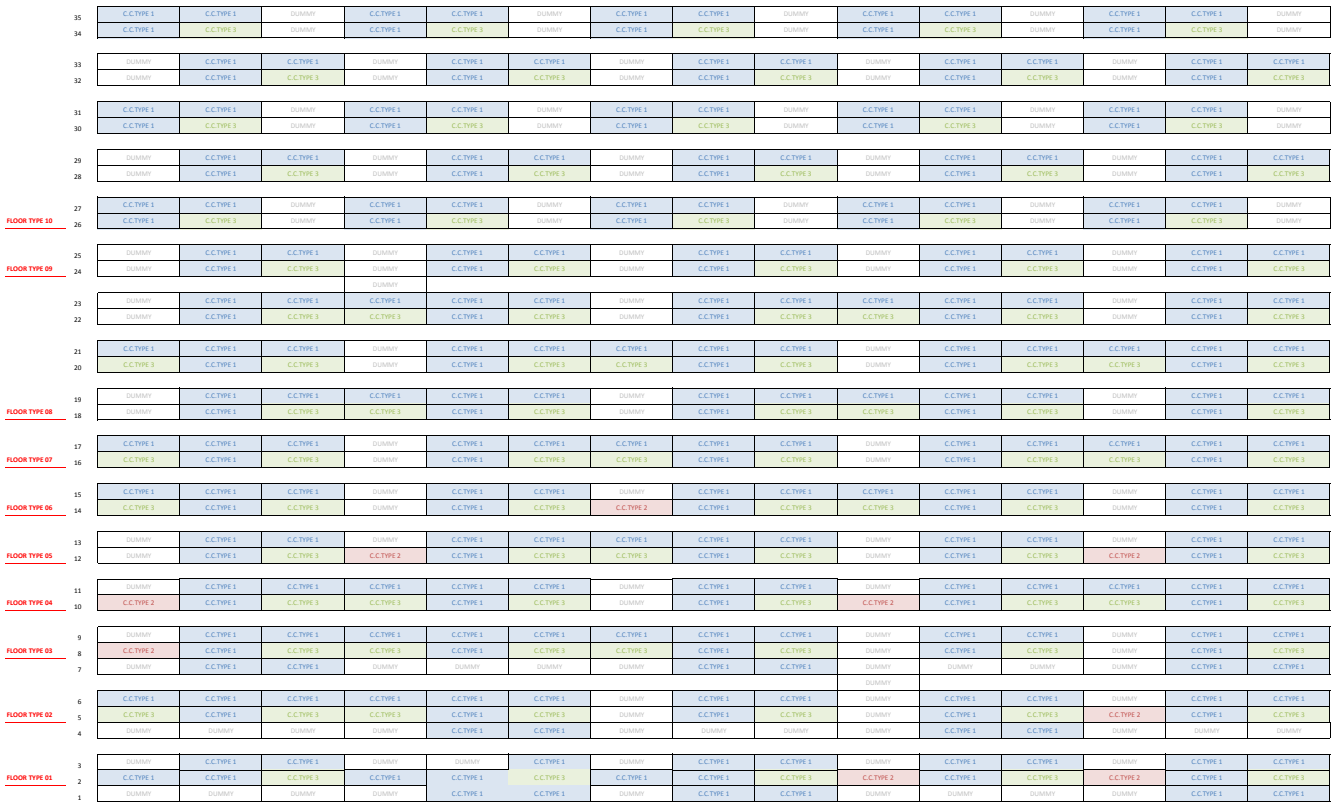
C.C.TYPE 1 | C.C.TYPE 1
C.C.TYPE 1 | C.C.TYPE 3 < ENT
C.C.TYPE 1 | C.C.TYPE 1

DUMMY TYPE

DUMMY



As discussed previously, each unit type is made up of different combinations of cell cluster types. By substituting unit types with cell cluster types, we can get an entire map of one of three cores. This also gives us ten different floor types. The rest of the floor types are repetitions of some of these ten.

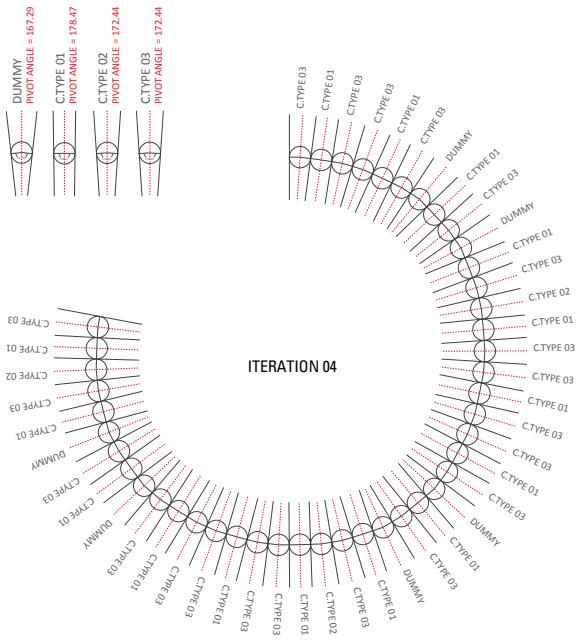


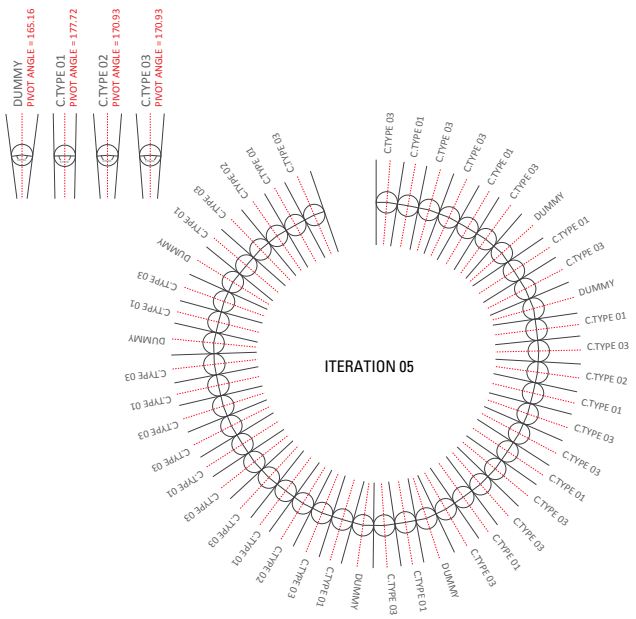
FLOOR TYPE 02
DUMMY
PIVOT ANGLE = 180.00
C.TYPE 01
PIVOT ANGLE = 180.00
C.TYPE 02
PIVOT ANGLE = 180.00
C.TYPE 03
PIVOT ANGLE = 180.00
INITIAL CONDITION
DUMMY
PIVOT ANGLE = 179.51
C.TYPE 01
PIVOT ANGLE = 179.76
C.TYPE 02
PIVOT ANGLE = 178.92
C.TYPE 03
PIVOT ANGLE = 178.92
DUMMY
PIVOT ANGLE = 179.00
C.TYPE 01
PIVOT ANGLE = 179.00
C.TYPE 02
PIVOT ANGLE = 177.87
C.TYPE 03
PIVOT ANGLE = 177.87
ITERATION 01
DUMMY
PIVOT ANGLE = 178.61
C.TYPE 01
PIVOT ANGLE = 178.88
C.TYPE 02
PIVOT ANGLE = 174.59
C.TYPE 03
PIVOT ANGLE = 174.59
ITERATION 02
ITERATION 03

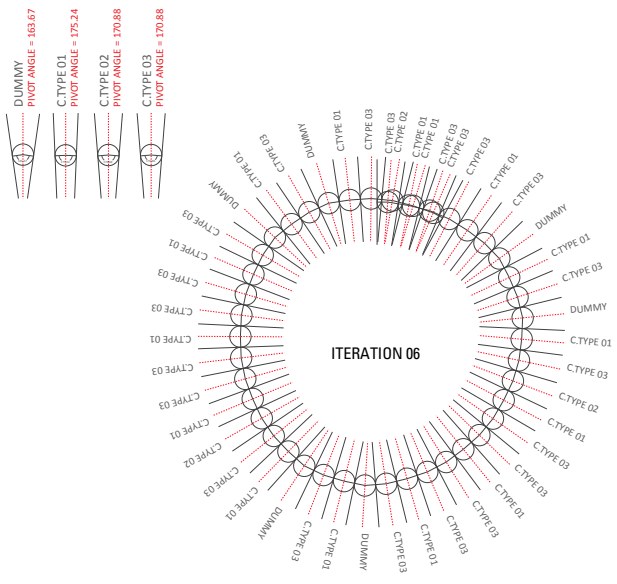
Now let's apply four cell cluster types to the floor types.

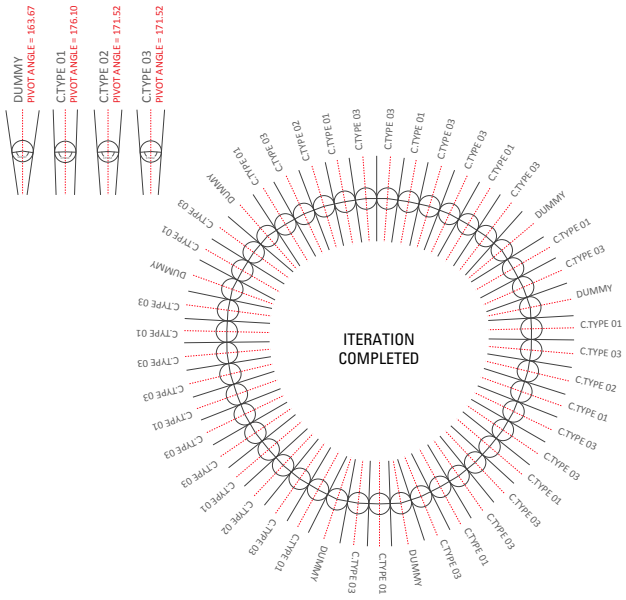
In the first step, we arrange them loosely as they are. For clarity purposes, we forget about the rest of cells for now, except for the large one that comes with pivotal points at the center.

In the step two, by changing the angle of the boundary lines, we find the optimal solution for packing entire cell cluster types. Diagrams are iteration process done in Grasshopper plug-in of Rhino 3d, to find optimal angles for all cell cluster types through series of trial-error method.

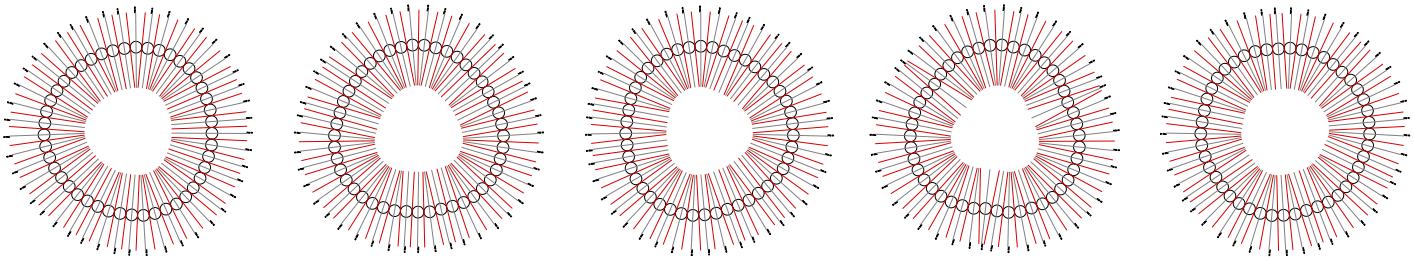




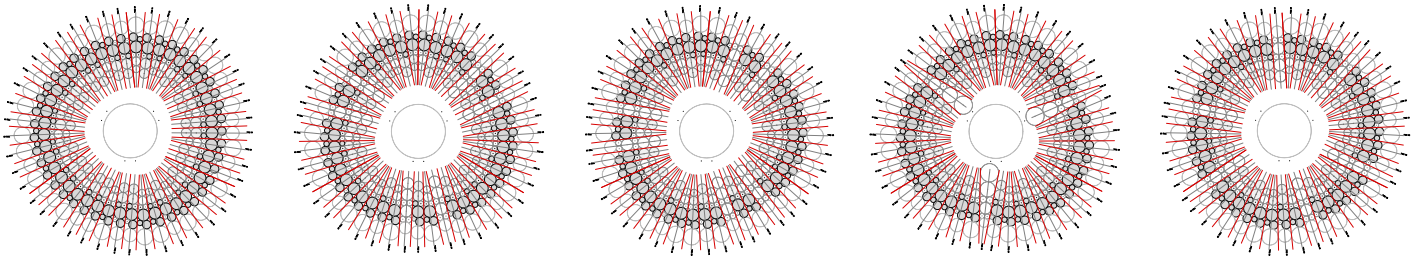




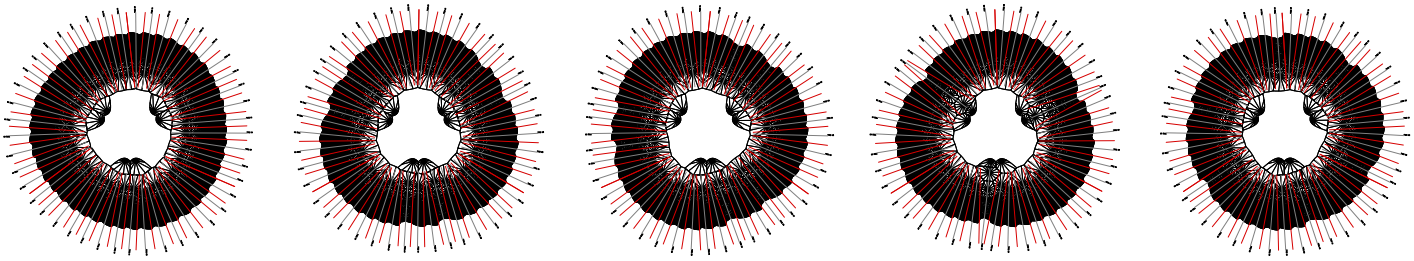
TEN FLOOR TYPES WITH PIVOTAL CELLS ONLY.



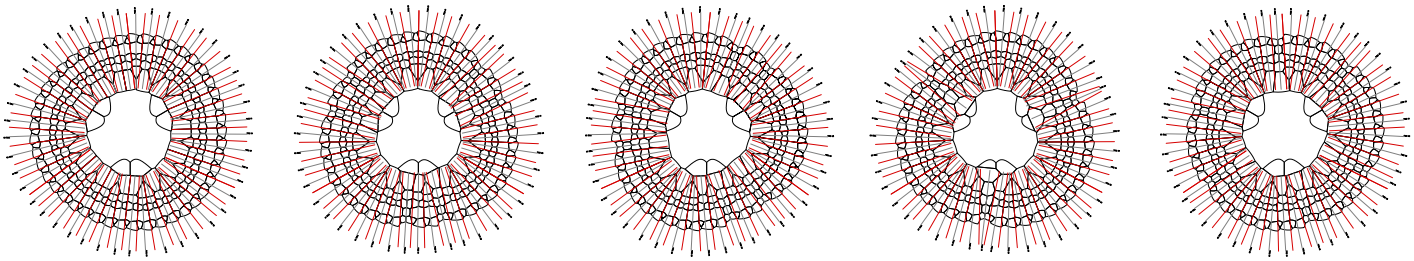
TEN FLOOR TYPES WITH ALL CELLS INCLUDED.

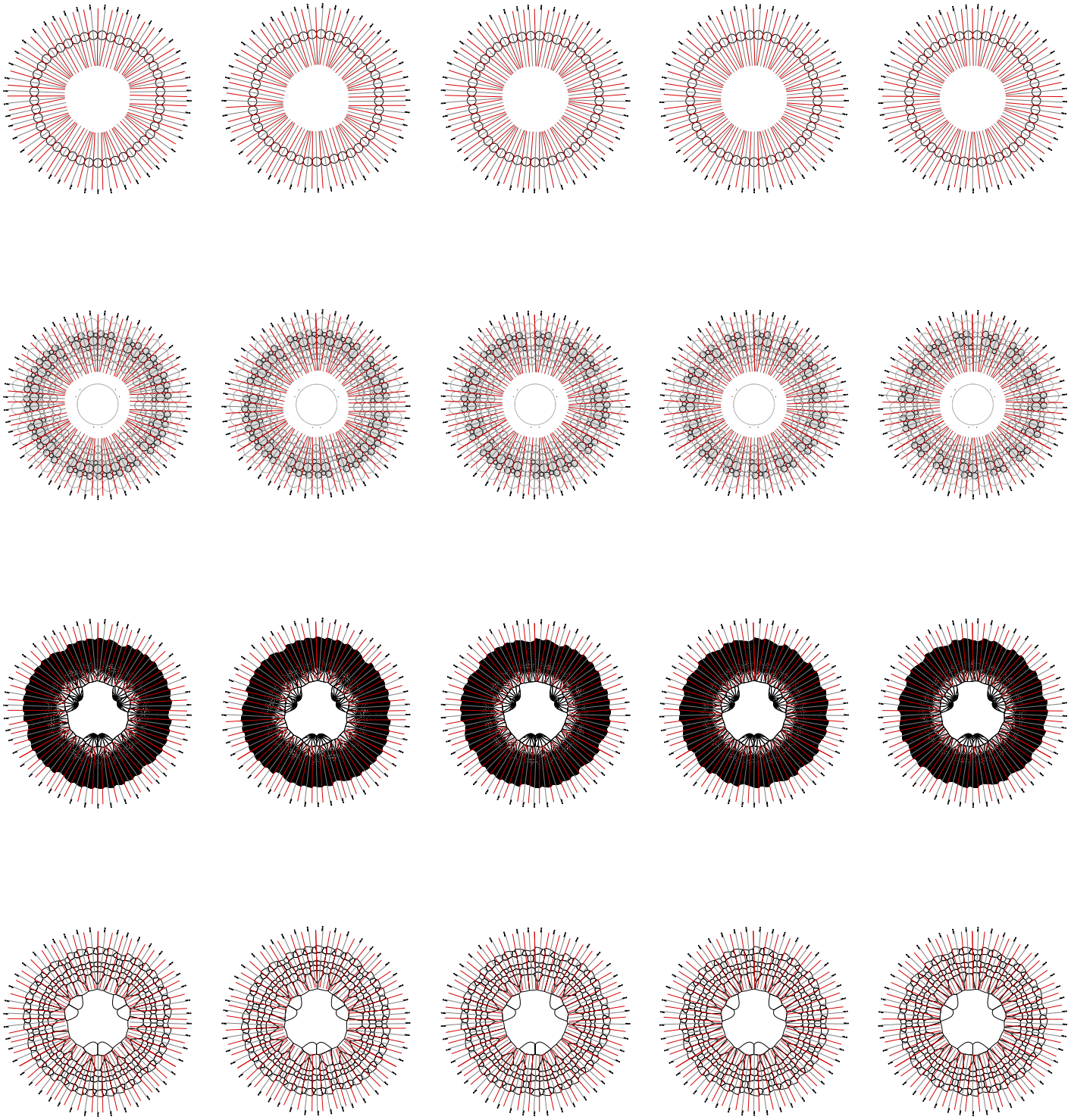


TEN FLOOR TYPES WITH A VORONOI CIRCLE SET BOUNDARY.

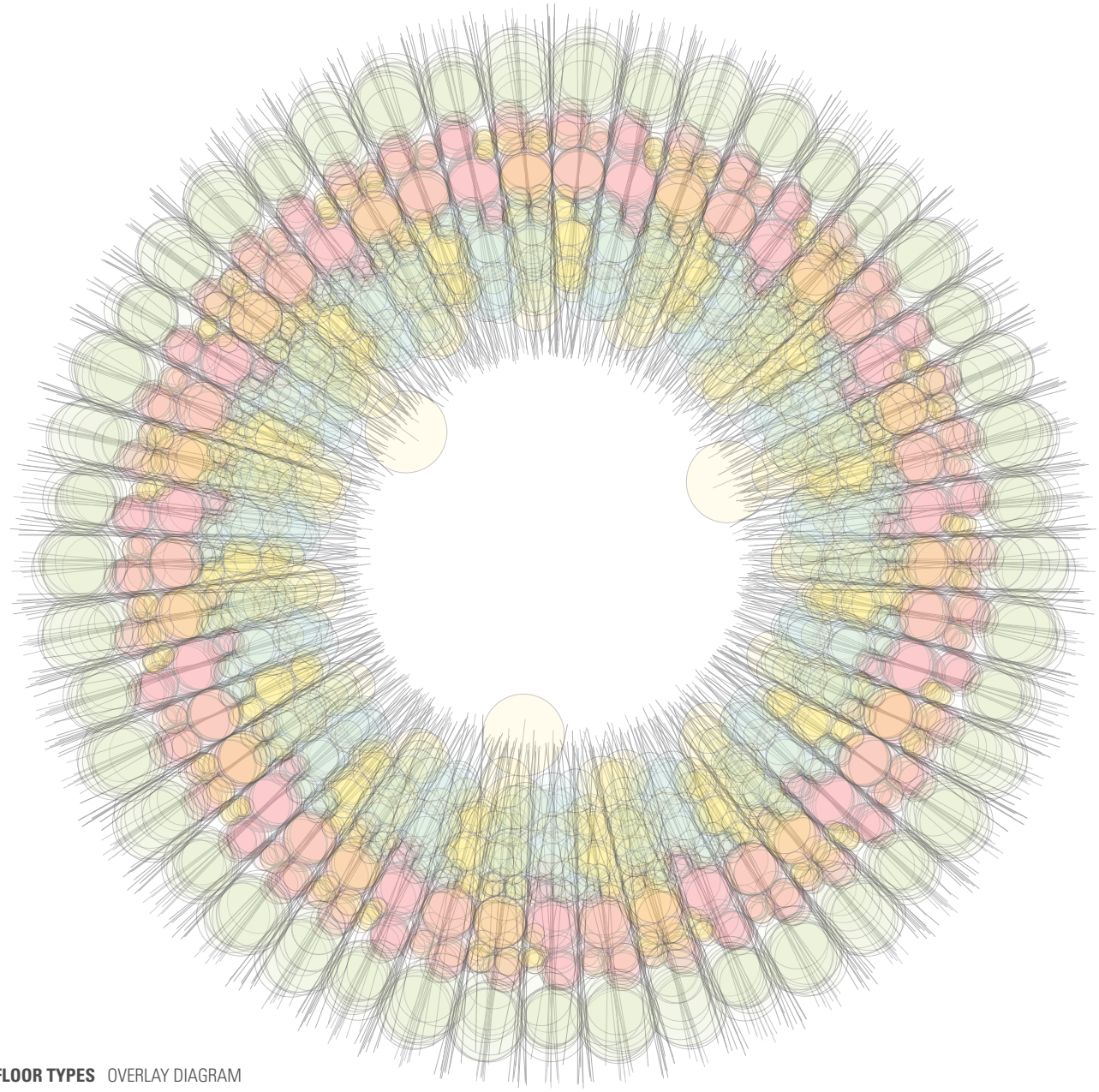


TEN FLOOR TYPES

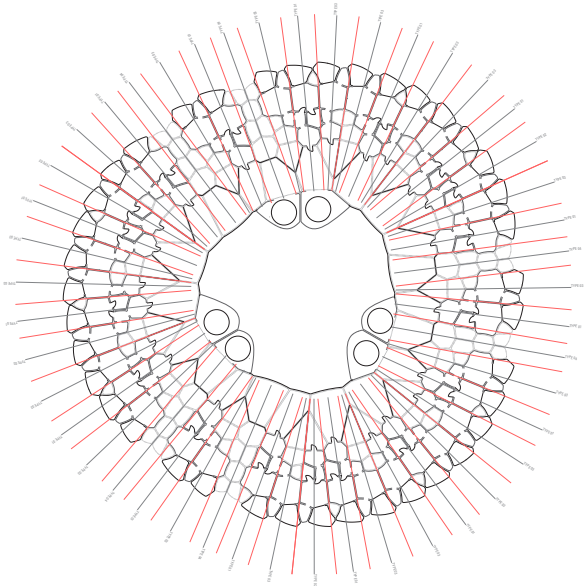




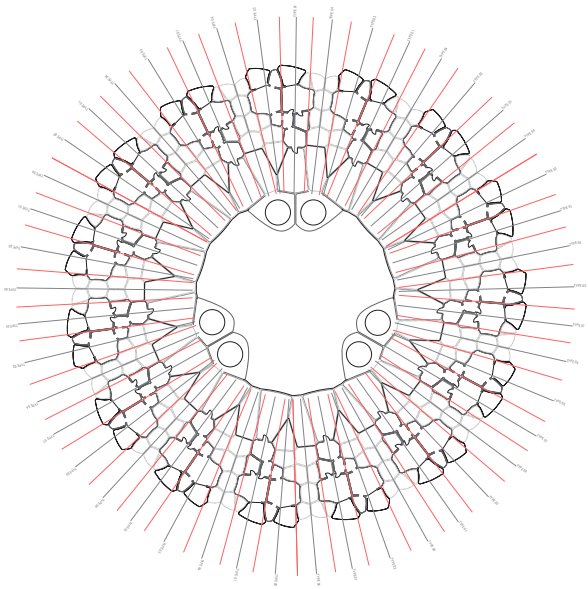
CONCLUSION



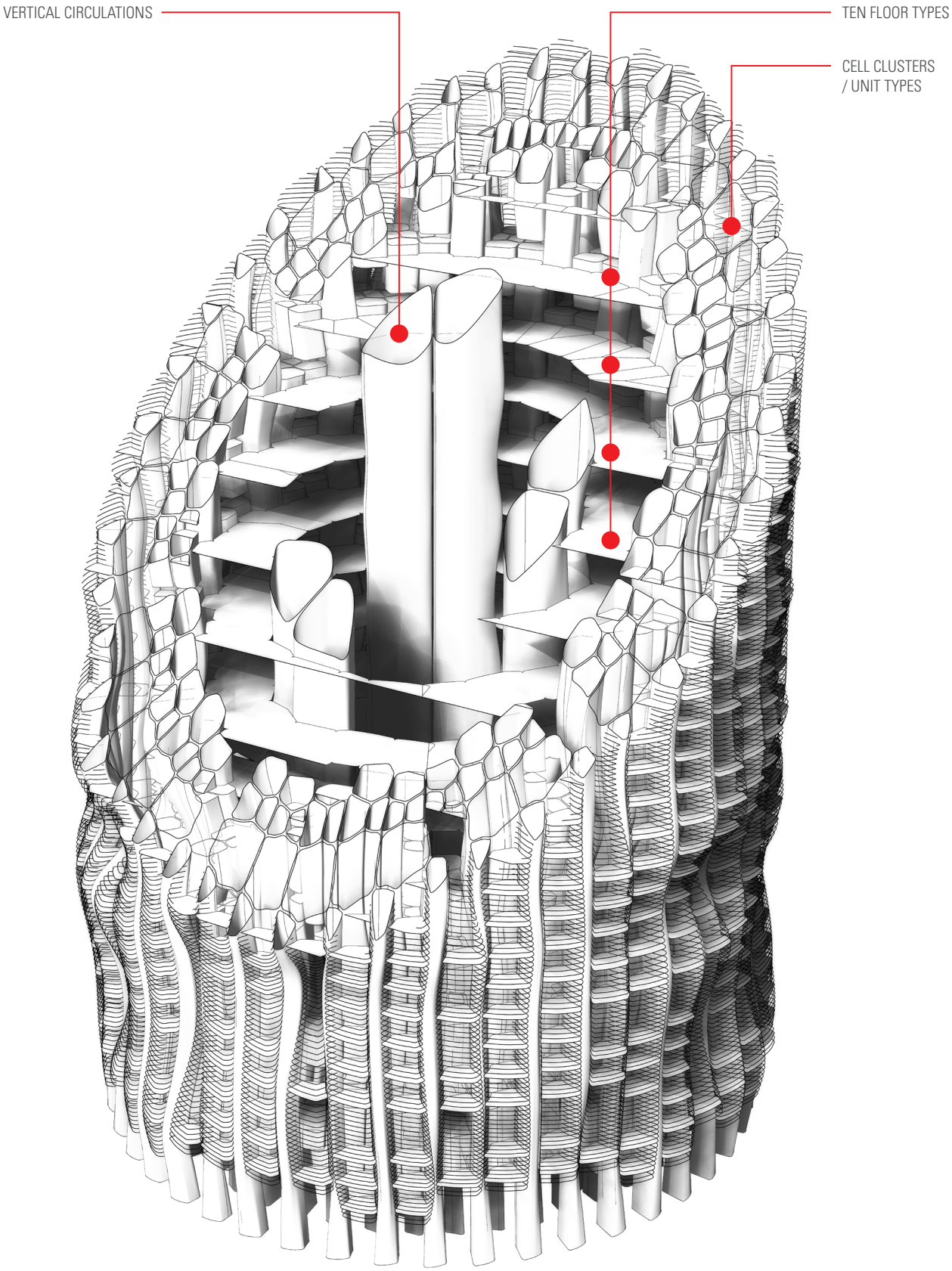
TEN FLOOR TYPES OVERLAY DIAGRAM

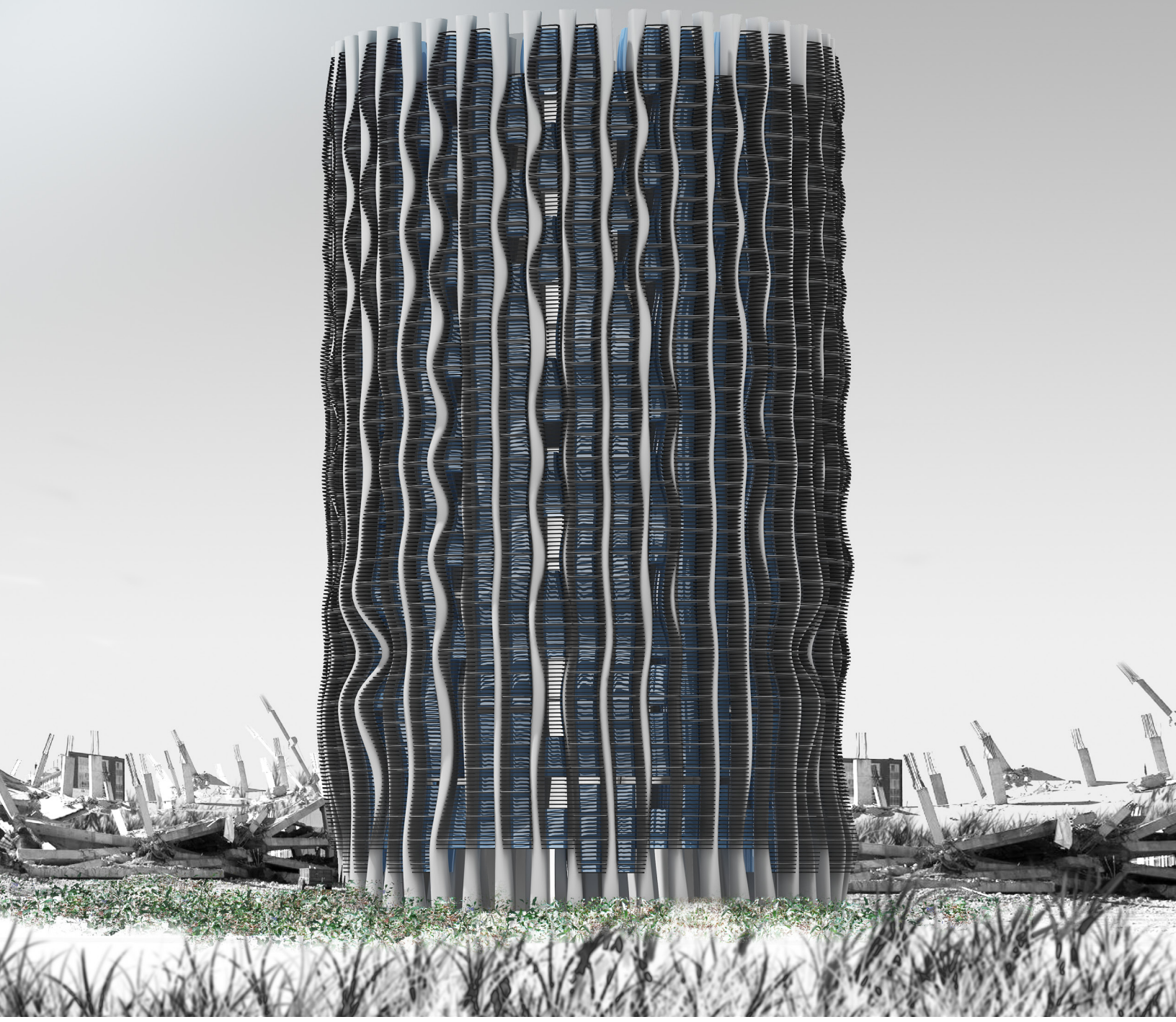


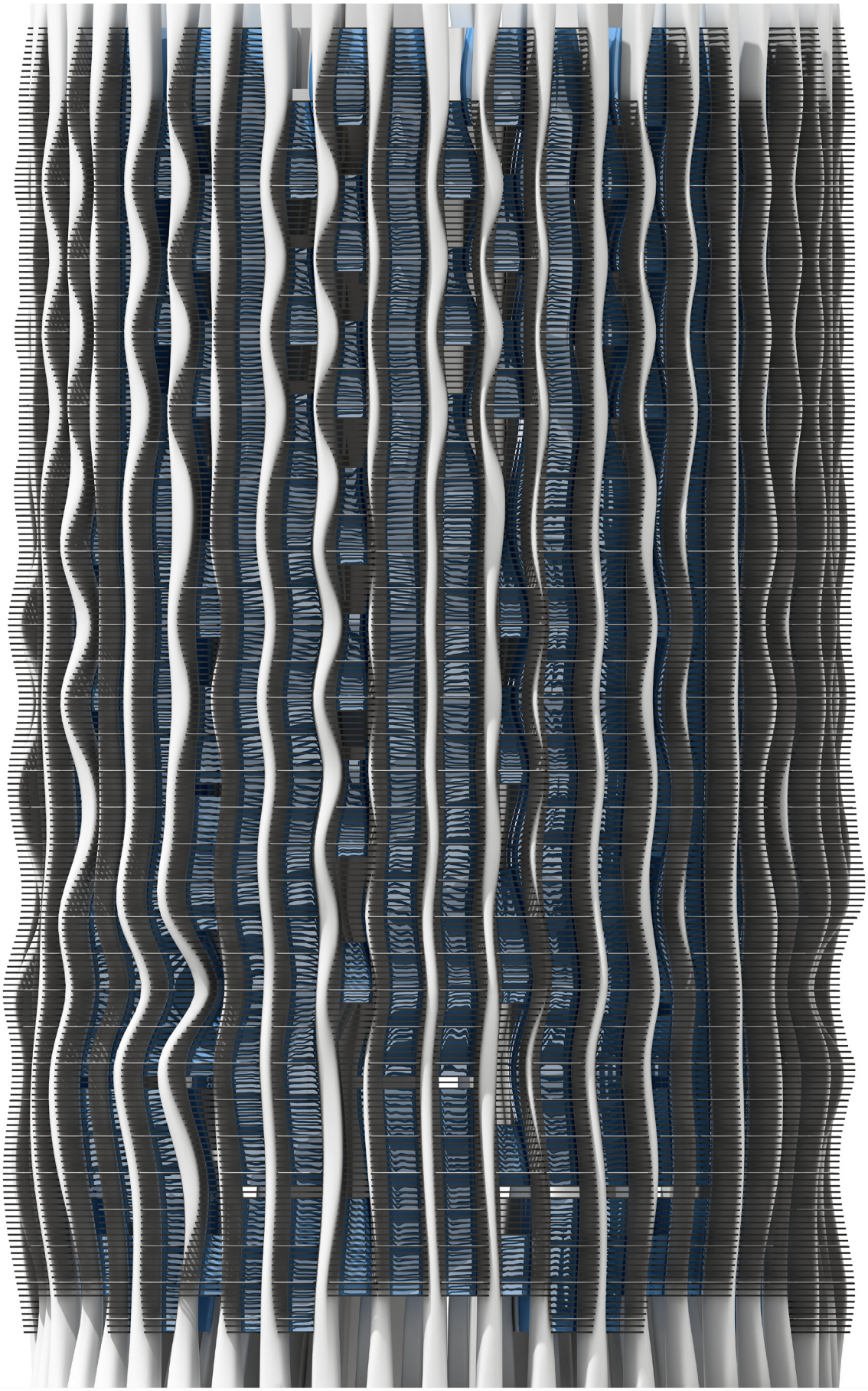
FLOOR TYPES 03 PLAN

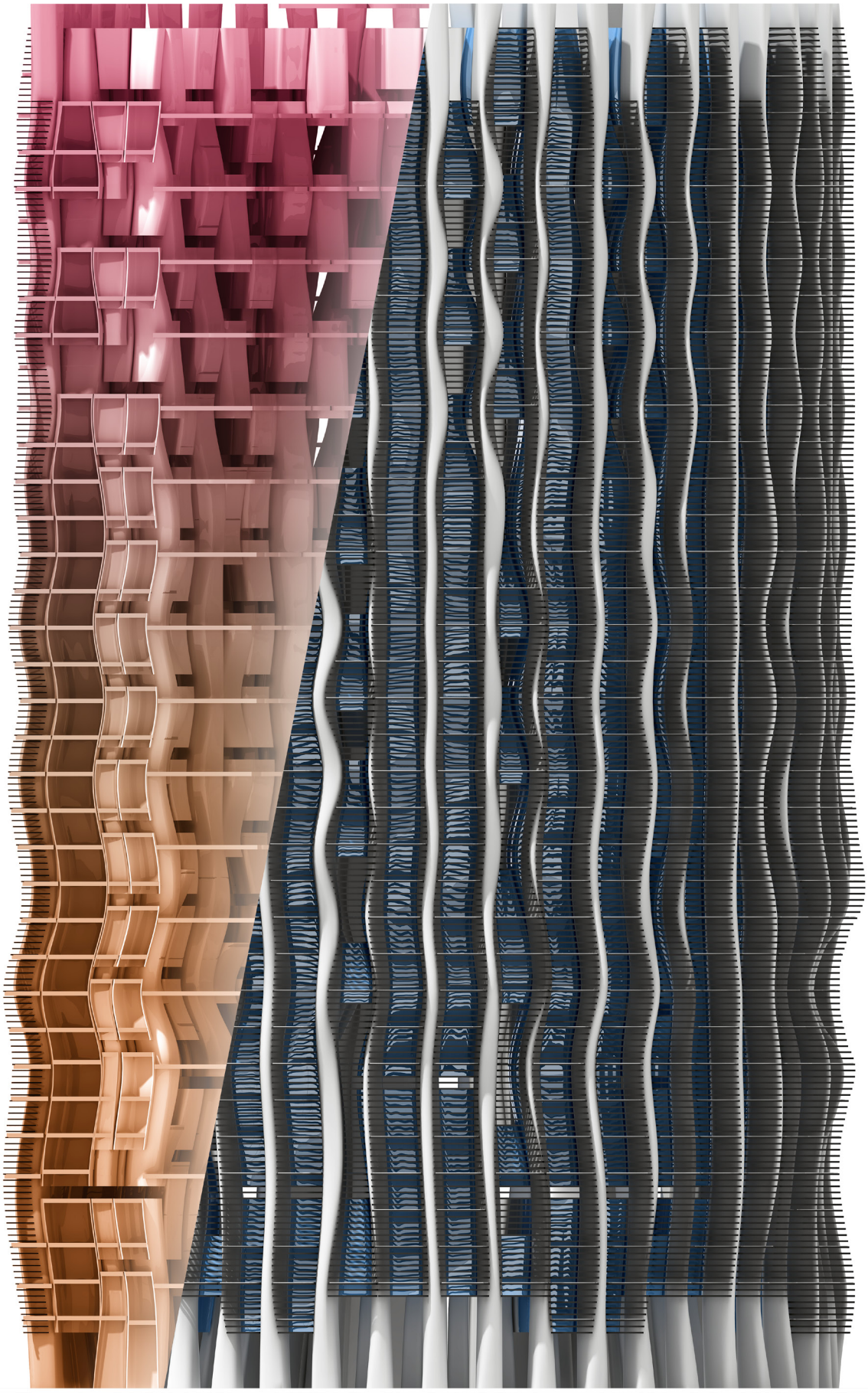


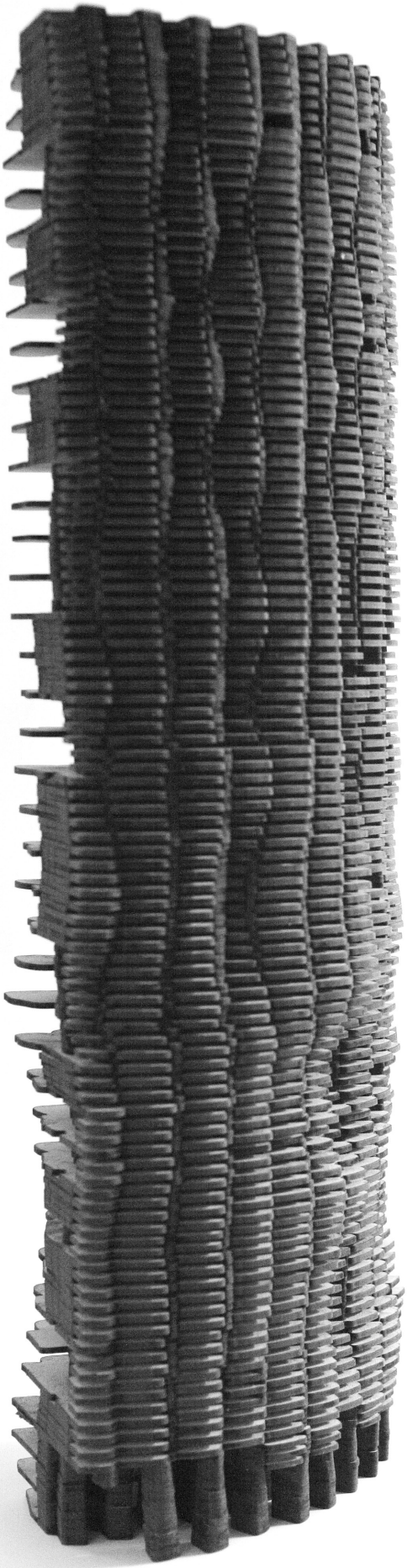
FLOOR TYPES 08 PLAN

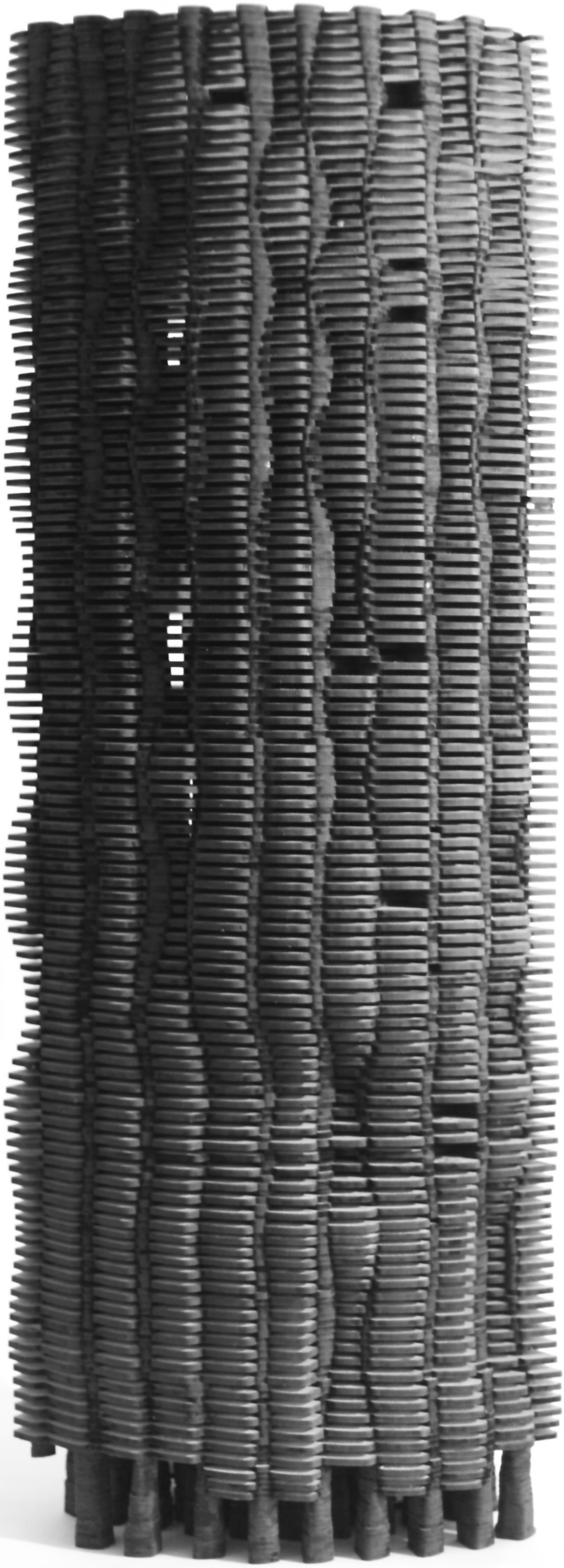

VERTICAL CIRCULATIONS
TEN FLOOR TYPES
CELL CLUSTERS
/ UNIT TYPES









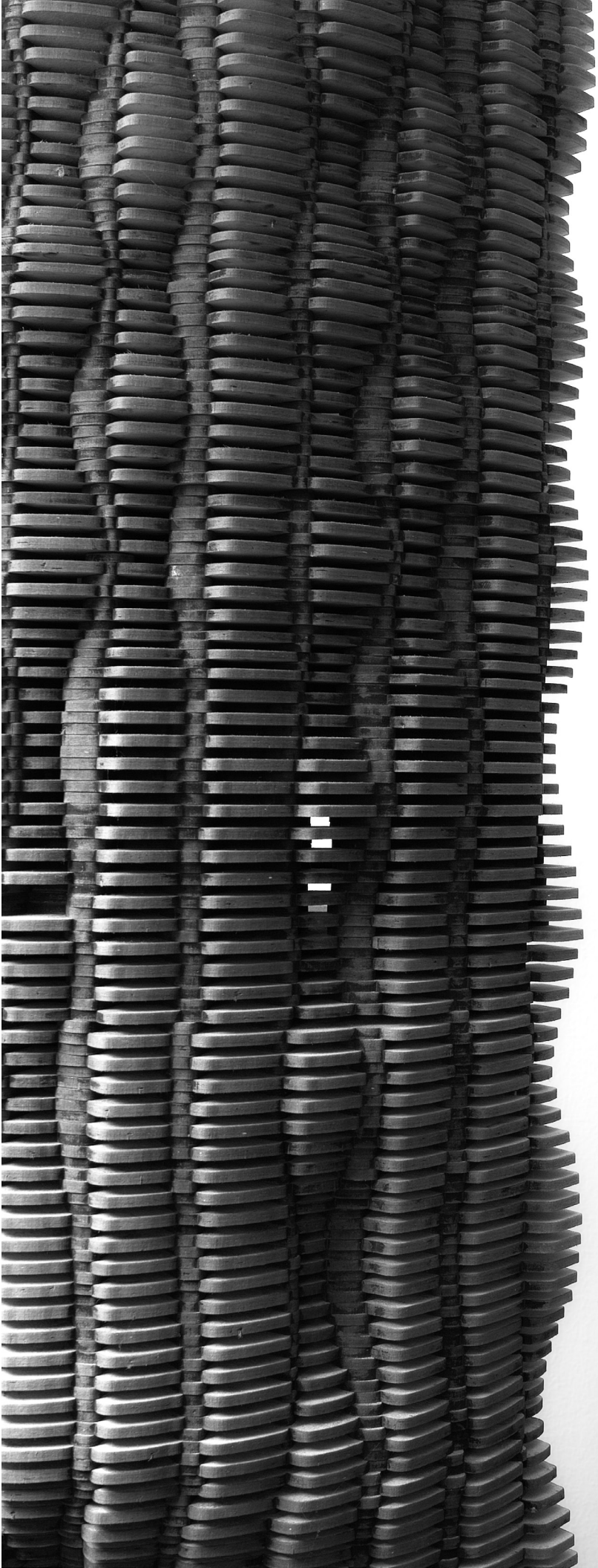


This project was initiated by a simple question: why are most of the walls/façades around us perpendicular to the floor? There might be pros and cons, but as is clearly investigated in "The Oblique Function" by Claude Parent and Paul Virilio, we could also use them in different ways, unless they are perpendicular to the floor plate. From the history of conflict amongst walls, façade, or cladding versus the structural skeleton described in Mohsen Mostafavi's Surface Architecture, our era has already witnessed the separation of walls/façade from the overall structure of a building. Although there might be technical / economic difficulties in actively designing and building non-vertical walls, it seemed to me that there must also be something profound behind the process that forces us to stick with rectilinear extrusion building types.

So I traced back the history of architecture to the point when wall / façade did not have to work as load-bearing element, and that brought me to the era of the Domino system proposed by Le Corbusier. From that point onward, no radical change has been made from a structural point of view. Although this is somehow tied up with "International Style," as its name implies, the style is rather a universal logic that is widely adaptable to site-specific conditions. In that sense, the style that came with the Domino system was to provide a flexible framework allowing architects greater design freedom, rather than functioning as a formal language associated with structural ideas, as the pre-modern styles did. Le Corbusier's Five Points of Architecture constitute another clear message that the intention of the Domino system was to encourage architects' free will in their design processes.

However in reality, the system, which was supposed to be a flexible framework, does not work as planned. Instead, it functions more as a formal style made out of a monotonous series of slabs and columns, as we can easily see from our built environment. This misappropriation—combined with other reasons such as material, building techniques, as well as cost—is actually holding back architects from moving on creatively. Then why not remove this rigid system from our buildings and see how that works?

Through the prototyping process, I proposed a flexible point grid system with various sizes of program bubbles, including dummy cells, which in turn creates an organic building shape while maintaining the rational logic of the Domino system as a virtual framework. Although I used the prototype to reorganize the Unite d'Habitation based on its existing inner logic, I suppose it is also capable of accommodating other logic from outside of architecture that might serve as a malleable framework.



References

Surface Architecture by David Leatherbarrow and Mohsen Mostafavi

Le Corbusier, L'Unite D'Habitation De Marseille by Jacques Sbriglio

The Adaptable House by Avi Friedman

Spatial Synthesis in Computer-Aided Building Design by Charles N. Eastman

Algorithms and Computation, Lecture Notes in Computer Sceience 1969

Architectural Morphology by Philip Steadman

Shape as Memory by Michael Leyton

Paul Virilio by Ian James

Re Constructing Architecture by Thomas A. Dutton

Objectile by Bernard Cache + Patrick Beaucé

Modern Housing Prototypes by Roger Sherwood

On Growth and Form by D'Arcy Wentworth Thompson

The Paul Virilio Reader by Steve Redhead

Softspace by Sean Lally and Jessica Young

Super - Recursive Algorithms by Mark Burgin