Master planning in semiconductor manufacturing – exercise

Outline of the LP model for master planning

We consider a semiconductor manufacturer with a three-stage production: Wafer fab, assembly, testing facility. In between are the respective inventories (Fig. 1).

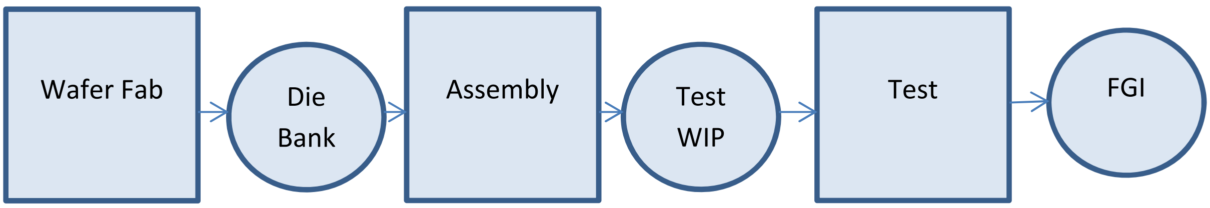

Figure 1: The three-stage production system (FGI … Finished Goods Inventory)

We set the following assumptions:

- The system produces one product (integrated circuit (IC), denoted *j*) and one wafer type (denoted *k*). Note that the product index changes from the wafer to the individual chip, termed *die*. Separating the dies that are produced on one wafer is performed in the assembly stage.
- The lead time for the wafer fab is 3 periods, the lead times for assembly and test are 1 period.
- The yield of assembly and test is 100% (no defective parts at these stages).

Decision variables:

P_{jt}^{A}, P_{jt}^{T}	Production quantity of product j in period t in the assembly and testing facility, respectively
P_{kt}^{W}	Production quantity of the wafer type k in period t in the wafer fab
R_{kt}^{W}	Release quantity of the wafer type k in period t to the wafer fab
I_{jt}^{A}, I_{jt}^{T}	Inventory of product j at the end of period t in the test WIP (after assembly) and in the FGI (after testing), respectively
I_{kt}^{W}	Inventory of the wafer type k at the end of period t

Parameters

D_{jt} Demand for product j in period t

f_{kj} Bill-of-Material coefficient, that is, the reciprocal of the number of dies meeting the quality standards (that can be assembled) per wafer. We assume 400 dies per wafer, that is, $f_{kj} = 1/400$.

The variables define the flows and the inventory levels in the system over time and thus capture the state of the entire system over time. Flows are defined for each edge (manufacturing plants, transportation links, etc.), the inventory levels are defined for each stock point.

The material balance equations define the network structure. Their common structure:

$$Inventory_t = Inventory_{t-1} + Input_t - Output_t \qquad \text{for all t}$$

For our case this means:

FGI balance equations:

$$I_{jt}^T = I_{j,t-1}^T + P_{jt}^T - D_{jt} \qquad \text{for all t}$$

Balance equations for test WIP:

$$I_{jt}^A = I_{j,t-1}^A + P_{jt}^A - P_{j,t+1}^T \qquad \text{for all t}$$

Note the lead time of 1 period for testing! The output from the test WIP in period t is the quantity finished from testing in period t+1.

Balance equations for wafers:

$$I_{kt}^W = I_{\mathrm{k},t-1}^W + P_{kt}^W - f_{kj} P_{j,t+1}^A \qquad \text{for all t}$$

Again the inventory is depleted by the assembly quantity of the next period. If 1 unit is assembled, the wafer inventory is reduced by f_{kj} units (which is less than 1 in this case; in the automotive industry when k denotes the wheels and j denotes the cars, f_{kj} would be 4).

Release quantities to the wafer fab (=output from the raw wafer inventory)

$$R_{kt}^W = P_{k,t+L}^W \qquad \text{for all t}$$

The following additional information is available:

Planning horizon: 12 periods.

Demand for the periods 1 to 12 in units (ICs):

10000,9000,8500,8000,9500,12000,14000,12000,12000,11500,10500,10000;

Available capacity per period: 13000 units in Test, 12000 units in assembly, 27 wafers in the fab. No additional capacity (overtime, etc.) is available.

Holding cost rates per unit and period: 4 and 5 money units (MUs) for Test WIP and FGI, respectively; 1200 MUs for the wafers.

Initial inventories: 2000 units in FGI, 4000 units in Test WIP, 100 Wafers.

We assume that work-in-process (WIP) in the fab at the start of the planning horizon is included in the initial wafer inventory, hence the production quantities in the fab for the periods 1 to 3 are zero (3 periods lead time!)

Exercise

Try to find a “good” (ideally: the optimal) production plan for the periods 1 to 12, that is, the production quantities and the inventory levels for all production stages (fab, assembly, test) and all inventories, respectively. You can try manually using a spreadsheet table, or optimize the master plan by linear programming.

Solution

We use the notation given above with some obvious modifications due to the LINDO syntax.

Model programmed in LINGO

```
Model:
Sets:
Period /1..12/:CT,CA,CW; !Capacities Test, Assembyl, Fab.;
Product /1..1/:IInitT,IInitA,IInitW;   ! initial inventories and holding
cost rates;
ProdPer (Product,Period): PA,PT,PW,IA,IT,IW,D,hT,hA,hW;  !Production
quantities and inventories in Test, Assembly, Fab, Demand;
Endsets

! Objective Function;
MIN = @Sum(ProdPer: hW * IW) + @Sum(ProdPer: hA * IA) + @Sum(ProdPer: hT *
IT);

!Inventory Balance Equations Wafers;
!For simplicity we assign the product intex J to the wafers as well;
@for(ProdPer(J,T) | T #NE# 1 #AND# T #LE# 9:
IW(J,T) - IW(J,T-1) - PW(J,T)  + F * PA(J,T+3)   = 0);
@for(ProdPer(J,T) | T #EQ# 1:
IW(J,T) - IInitW(J) - PW(J,T)  + F * PA(J,T+3)   = 0);

!Inventory Balance Equations Test WIP;
@for(ProdPer(J,T) | T #NE# 1 #AND# T #LE# 11:
IA(J,T) - IA(J,T-1) - PA(J,T)  + PT(J,T+1)   = 0);
@for(ProdPer(J,T) | T #EQ# 1:
IA(J,T) - IInitA(J) - PA(J,T)  + PT(J,T+1)   = 0);

!Inventory Balance Equations FGI;
@for(ProdPer(J,T) | T #NE# 1:
IT(J,T) - IT(J,T-1) - PT(J,T)  + D(J,T)   = 0);
@for(ProdPer(J,T) | T #EQ# 1:
IT(J,T) - IInitT(J) - PT(J,T)  + D(J,T)   = 0);


!Capacity Constraints;
@for(Period(T):
@Sum(Product(J): PT(J,T))  < CT(T));
@for(Period(T):
@Sum(Product(J): PA(J,T))  < CA(T));
@for(Period(T):
@Sum(Product(J): PW(J,T))  < CW(T));


!Definition Regular Capacity;
@for(Period(T):
CT(T) = 13000;
CA(T) = 12000;
CW(T) = 27);


Data:
D=10000,9000,8500,8000,9500,12000,14000,12000,12000,11500,10500,10000;
!Initial inventories;
IInitT=2000;
IInitA=4000;
IInitW=100;
```

```
F=0.0025; ! Number of wafers for one integratred curcuit;
!Holding cost coefficients;
hW=1200,1200,1200,1200,1200,1200,1200,1200,1200,1200,1200,1200;
hA=4,4,4,4,4,4,4,4,4,4,4,4;
hT=5,5,5,5,5,5,5,5,5,5,5,5;

Enddata

End
```

Model formulation generated by LINGO

```
  MODEL:
   [_1] MIN= 4 * IA_1_1 + 5 * IT_1_1 + 1200 * IW_1_1 + 4 * IA_1_2 + 5 *
   IT_1_2 + 1200 * IW_1_2 + 4 * IA_1_3 + 5 * IT_1_3 + 1200 * IW_1_3 + 4 *
   IA_1_4 + 5 * IT_1_4 + 1200 * IW_1_4 + 4 * IA_1_5 + 5 * IT_1_5 + 1200 *
   IW_1_5 + 4 * IA_1_6 + 5 * IT_1_6 + 1200 * IW_1_6 + 4 * IA_1_7 + 5 *
   IT_1_7 + 1200 * IW_1_7 + 4 * IA_1_8 + 5 * IT_1_8 + 1200 * IW_1_8 + 4 *
   IA_1_9 + 5 * IT_1_9 + 1200 * IW_1_9 + 4 * IA_1_10 + 5 * IT_1_10 + 1200 *
   IW_1_10 + 4 * IA_1_11 + 5 * IT_1_11 + 1200 * IW_1_11 + 4 * IA_1_12 + 5 *
   IT_1_12 + 1200 * IW_1_12 ;
   [_2] - IW_1_1 - PW_1_2 + IW_1_2 + 0.0025 * PA_1_5 = 0 ;
   [_3] - IW_1_2 - PW_1_3 + IW_1_3 + 0.0025 * PA_1_6 = 0 ;
   [_4] - IW_1_3 - PW_1_4 + IW_1_4 + 0.0025 * PA_1_7 = 0 ;
   [_5] - IW_1_4 - PW_1_5 + IW_1_5 + 0.0025 * PA_1_8 = 0 ;
   [_6] - IW_1_5 - PW_1_6 + IW_1_6 + 0.0025 * PA_1_9 = 0 ;
   [_7] - IW_1_6 - PW_1_7 + IW_1_7 + 0.0025 * PA_1_10 = 0 ;
   [_8] - IW_1_7 - PW_1_8 + IW_1_8 + 0.0025 * PA_1_11 = 0 ;
   [_9] - IW_1_8 - PW_1_9 + IW_1_9 + 0.0025 * PA_1_12 = 0 ;
   [_10] - PW_1_1 + IW_1_1 + 0.0025 * PA_1_4 = 100 ;
   [_11] - IA_1_1 - PA_1_2 + IA_1_2 + PT_1_3 = 0 ;
   [_12] - IA_1_2 - PA_1_3 + IA_1_3 + PT_1_4 = 0 ;
   [_13] - IA_1_3 - PA_1_4 + IA_1_4 + PT_1_5 = 0 ;
   [_14] - IA_1_4 - PA_1_5 + IA_1_5 + PT_1_6 = 0 ;
   [_15] - IA_1_5 - PA_1_6 + IA_1_6 + PT_1_7 = 0 ;
   [_16] - IA_1_6 - PA_1_7 + IA_1_7 + PT_1_8 = 0 ;
   [_17] - IA_1_7 - PA_1_8 + IA_1_8 + PT_1_9 = 0 ;
   [_18] - IA_1_8 - PA_1_9 + IA_1_9 + PT_1_10 = 0 ;
   [_19] - IA_1_9 - PA_1_10 + IA_1_10 + PT_1_11 = 0 ;
   [_20] - IA_1_10 - PA_1_11 + IA_1_11 + PT_1_12 = 0 ;
   [_21] - PA_1_1 + IA_1_1 + PT_1_2 = 4000 ;
   [_22] - IT_1_1 - PT_1_2 + IT_1_2 = - 9000 ;
   [_23] - IT_1_2 - PT_1_3 + IT_1_3 = - 8500 ;
   [_24] - IT_1_3 - PT_1_4 + IT_1_4 = - 8000 ;
   [_25] - IT_1_4 - PT_1_5 + IT_1_5 = - 9500 ;
   [_26] - IT_1_5 - PT_1_6 + IT_1_6 = - 12000 ;
   [_27] - IT_1_6 - PT_1_7 + IT_1_7 = - 14000 ;
   [_28] - IT_1_7 - PT_1_8 + IT_1_8 = - 12000 ;
   [_29] - IT_1_8 - PT_1_9 + IT_1_9 = - 12000 ;
   [_30] - IT_1_9 - PT_1_10 + IT_1_10 = - 11500 ;
   [_31] - IT_1_10 - PT_1_11 + IT_1_11 = - 10500 ;
   [_32] - IT_1_11 - PT_1_12 + IT_1_12 = - 10000 ;
   [_33] - PT_1_1 + IT_1_1 = - 8000 ;
   [_34] PT_1_1 <= 13000 ;
   [_35] PT_1_2 <= 13000 ;
   [_36] PT_1_3 <= 13000 ;
   [_37] PT_1_4 <= 13000 ;
   [_38] PT_1_5 <= 13000 ;
   [_39] PT_1_6 <= 13000 ;
   [_40] PT_1_7 <= 13000 ;
   [_41] PT_1_8 <= 13000 ;
```

```
 [_42] PT_1_9 <= 13000 ;
 [_43] PT_1_10 <= 13000 ;
 [_44] PT_1_11 <= 13000 ;
 [_45] PT_1_12 <= 13000 ;
 [_46] PA_1_1 <= 12000 ;
 [_47] PA_1_2 <= 12000 ;
 [_48] PA_1_3 <= 12000 ;
 [_49] PA_1_4 <= 12000 ;
 [_50] PA_1_5 <= 12000 ;
 [_51] PA_1_6 <= 12000 ;
 [_52] PA_1_7 <= 12000 ;
 [_53] PA_1_8 <= 12000 ;
 [_54] PA_1_9 <= 12000 ;
 [_55] PA_1_10 <= 12000 ;
 [_56] PA_1_11 <= 12000 ;
 [_57] PA_1_12 <= 12000 ;
 [_58] PW_1_1 <= 27 ;
 [_59] PW_1_2 <= 27 ;
 [_60] PW_1_3 <= 27 ;
 [_61] PW_1_4 <= 27 ;
 [_62] PW_1_5 <= 27 ;
 [_63] PW_1_6 <= 27 ;
 [_64] PW_1_7 <= 27 ;
 [_65] PW_1_8 <= 27 ;
 [_66] PW_1_9 <= 27 ;
 [_67] PW_1_10 <= 27 ;
 [_68] PW_1_11 <= 27 ;
 [_69] PW_1_12 <= 27 ;
END
```

Model Solution

```
Global optimal solution found.
Objective value:                              173300.0
Infeasibilities:                              0.000000
Total solver iterations:                            26
```

Variable	Value	Reduced Cost
F	0.2500000E-02	0.000000
CT(1)	13000.00	0.000000
CT(2)	13000.00	0.000000
CT(3)	13000.00	0.000000
CT(4)	13000.00	0.000000
CT(5)	13000.00	0.000000
CT(6)	13000.00	0.000000
CT(7)	13000.00	0.000000
CT(8)	13000.00	0.000000
CT(9)	13000.00	0.000000
CT(10)	13000.00	0.000000
CT(11)	13000.00	0.000000
CT(12)	13000.00	0.000000
CA(1)	12000.00	0.000000
CA(2)	12000.00	0.000000
CA(3)	12000.00	0.000000
CA(4)	12000.00	0.000000
CA(5)	12000.00	0.000000
CA(6)	12000.00	0.000000
CA(7)	12000.00	0.000000
CA(8)	12000.00	0.000000
CA(9)	12000.00	0.000000
CA(10)	12000.00	0.000000
CA(11)	12000.00	0.000000
CA(12)	12000.00	0.000000
CW(1)	27.00000	0.000000
CW(2)	27.00000	0.000000
CW(3)	27.00000	0.000000
CW(4)	27.00000	0.000000
CW(5)	27.00000	0.000000
CW(6)	27.00000	0.000000
CW(7)	27.00000	0.000000
CW(8)	27.00000	0.000000
CW(9)	27.00000	0.000000
CW(10)	27.00000	0.000000
CW(11)	27.00000	0.000000
CW(12)	27.00000	0.000000
IINITT(1)	2000.000	0.000000
IINITA(1)	4000.000	0.000000
IINITW(1)	100.0000	0.000000
PA(1, 1)	5000.000	0.000000
PA(1, 2)	8500.000	0.000000
PA(1, 3)	8000.000	0.000000
PA(1, 4)	11500.00	0.000000
PA(1, 5)	12000.00	0.000000
PA(1, 6)	12000.00	0.000000
PA(1, 7)	12000.00	0.000000
PA(1, 8)	12000.00	0.000000
PA(1, 9)	11500.00	0.000000
PA(1, 10)	10500.00	0.000000
PA(1, 11)	10000.00	0.000000
PA(1, 12)	10800.00	0.000000
PT(1, 1)	8000.000	0.000000
PT(1, 2)	9000.000	0.000000

PT(1, 3)	8500.000	0.000000
PT(1, 4)	8000.000	0.000000
PT(1, 5)	9500.000	0.000000
PT(1, 6)	13000.00	0.000000
PT(1, 7)	13000.00	0.000000
PT(1, 8)	12000.00	0.000000
PT(1, 9)	12000.00	0.000000
PT(1, 10)	11500.00	0.000000
PT(1, 11)	10500.00	0.000000
PT(1, 12)	10000.00	0.000000
PW(1, 1)	0.000000	3600.000
PW(1, 2)	0.000000	2400.000
PW(1, 3)	0.000000	1200.000
PW(1, 4)	23.50000	0.000000
PW(1, 5)	27.00000	0.000000
PW(1, 6)	27.00000	0.000000
PW(1, 7)	26.25000	0.000000
PW(1, 8)	25.00000	0.000000
PW(1, 9)	27.00000	0.000000
PW(1, 10)	0.000000	0.000000
PW(1, 11)	0.000000	0.000000
PW(1, 12)	0.000000	0.000000
IA(1, 1)	0.000000	4.000000
IA(1, 2)	0.000000	4.000000
IA(1, 3)	0.000000	13.00000
IA(1, 4)	2000.000	0.000000
IA(1, 5)	1000.000	0.000000
IA(1, 6)	0.000000	3.000000
IA(1, 7)	0.000000	1.000000
IA(1, 8)	0.000000	1.000000
IA(1, 9)	0.000000	10.00000
IA(1, 10)	0.000000	4.000000
IA(1, 11)	0.000000	4.000000
IA(1, 12)	0.000000	4.000000
IT(1, 1)	0.000000	5.000000
IT(1, 2)	0.000000	5.000000
IT(1, 3)	0.000000	5.000000
IT(1, 4)	0.000000	14.00000
IT(1, 5)	0.000000	0.000000
IT(1, 6)	1000.000	0.000000
IT(1, 7)	0.000000	6.000000
IT(1, 8)	0.000000	2.000000
IT(1, 9)	0.000000	2.000000
IT(1, 10)	0.000000	11.00000
IT(1, 11)	0.000000	5.000000
IT(1, 12)	0.000000	5.000000
IW(1, 1)	71.25000	0.000000
IW(1, 2)	41.25000	0.000000
IW(1, 3)	11.25000	0.000000
IW(1, 4)	4.750000	0.000000
IW(1, 5)	1.750000	0.000000
IW(1, 6)	0.000000	3600.000
IW(1, 7)	0.000000	1200.000
IW(1, 8)	0.000000	1200.000
IW(1, 9)	0.000000	1200.000
IW(1, 10)	0.000000	1200.000
IW(1, 11)	0.000000	1200.000
IW(1, 12)	0.000000	1200.000
D(1, 1)	10000.00	0.000000
D(1, 2)	9000.000	0.000000
D(1, 3)	8500.000	0.000000
D(1, 4)	8000.000	0.000000

D(1, 5)	9500.000	0.000000
D(1, 6)	12000.00	0.000000
D(1, 7)	14000.00	0.000000
D(1, 8)	12000.00	0.000000
D(1, 9)	12000.00	0.000000
D(1, 10)	11500.00	0.000000
D(1, 11)	10500.00	0.000000
D(1, 12)	10000.00	0.000000
HT(1, 1)	5.000000	0.000000
HT(1, 2)	5.000000	0.000000
HT(1, 3)	5.000000	0.000000
HT(1, 4)	5.000000	0.000000
HT(1, 5)	5.000000	0.000000
HT(1, 6)	5.000000	0.000000
HT(1, 7)	5.000000	0.000000
HT(1, 8)	5.000000	0.000000
HT(1, 9)	5.000000	0.000000
HT(1, 10)	5.000000	0.000000
HT(1, 11)	5.000000	0.000000
HT(1, 12)	5.000000	0.000000
HA(1, 1)	4.000000	0.000000
HA(1, 2)	4.000000	0.000000
HA(1, 3)	4.000000	0.000000
HA(1, 4)	4.000000	0.000000
HA(1, 5)	4.000000	0.000000
HA(1, 6)	4.000000	0.000000
HA(1, 7)	4.000000	0.000000
HA(1, 8)	4.000000	0.000000
HA(1, 9)	4.000000	0.000000
HA(1, 10)	4.000000	0.000000
HA(1, 11)	4.000000	0.000000
HA(1, 12)	4.000000	0.000000
HW(1, 1)	1200.000	0.000000
HW(1, 2)	1200.000	0.000000
HW(1, 3)	1200.000	0.000000
HW(1, 4)	1200.000	0.000000
HW(1, 5)	1200.000	0.000000
HW(1, 6)	1200.000	0.000000
HW(1, 7)	1200.000	0.000000
HW(1, 8)	1200.000	0.000000
HW(1, 9)	1200.000	0.000000
HW(1, 10)	1200.000	0.000000
HW(1, 11)	1200.000	0.000000
HW(1, 12)	1200.000	0.000000

Row	Slack or Surplus	Dual Price
1	173300.0	-1.000000
2	0.000000	-2400.000
3	0.000000	-1200.000
4	0.000000	0.000000
5	0.000000	1200.000
6	0.000000	2400.000
7	0.000000	0.000000
8	0.000000	0.000000
9	0.000000	0.000000
10	0.000000	-3600.000
11	0.000000	0.000000
12	0.000000	0.000000
13	0.000000	-9.000000
14	0.000000	-5.000000
15	0.000000	-1.000000

16	0.000000	0.000000
17	0.000000	3.000000
18	0.000000	6.000000
19	0.000000	0.000000
20	0.000000	0.000000
21	0.000000	0.000000
22	0.000000	0.000000
23	0.000000	0.000000
24	0.000000	0.000000
25	0.000000	-9.000000
26	0.000000	-4.000000
27	0.000000	1.000000
28	0.000000	0.000000
29	0.000000	3.000000
30	0.000000	6.000000
31	0.000000	0.000000
32	0.000000	0.000000
33	0.000000	0.000000
34	5000.000	0.000000
35	4000.000	0.000000
36	4500.000	0.000000
37	5000.000	0.000000
38	3500.000	0.000000
39	0.000000	1.000000
40	0.000000	2.000000
41	1000.000	0.000000
42	1000.000	0.000000
43	1500.000	0.000000
44	2500.000	0.000000
45	3000.000	0.000000
46	7000.000	0.000000
47	3500.000	0.000000
48	4000.000	0.000000
49	500.0000	0.000000
50	0.000000	1.000000
51	0.000000	2.000000
52	0.000000	0.000000
53	0.000000	0.000000
54	500.0000	0.000000
55	1500.000	0.000000
56	2000.000	0.000000
57	1200.000	0.000000
58	27.00000	0.000000
59	27.00000	0.000000
60	27.00000	0.000000
61	3.500000	0.000000
62	0.000000	1200.000
63	0.000000	2400.000
64	0.7500000	0.000000
65	2.000000	0.000000
66	0.000000	0.000000
67	27.00000	0.000000
68	27.00000	0.000000
69	27.00000	0.000000
70	0.000000	0.000000
71	0.000000	0.000000
72	0.000000	0.000000
73	0.000000	0.000000
74	0.000000	0.000000
75	0.000000	0.000000
76	0.000000	0.000000
77	0.000000	0.000000

78	0.000000	0.000000
79	0.000000	0.000000
80	0.000000	0.000000
81	0.000000	0.000000
82	0.000000	0.000000
83	0.000000	1.000000
84	0.000000	1200.000
85	0.000000	1.000000
86	0.000000	2.000000
87	0.000000	2400.000
88	0.000000	2.000000
89	0.000000	0.000000
90	0.000000	0.000000
91	0.000000	0.000000
92	0.000000	0.000000
93	0.000000	0.000000
94	0.000000	0.000000
95	0.000000	0.000000
96	0.000000	0.000000
97	0.000000	0.000000
98	0.000000	0.000000
99	0.000000	0.000000
100	0.000000	0.000000
101	0.000000	0.000000
102	0.000000	0.000000
103	0.000000	0.000000
104	0.000000	0.000000
105	0.000000	0.000000

The values of the decision variables in the optimal solution can be seen immediately, the interpretation of the other results (reduced cost, Slack or Surplus, Dual Price) is described in the exercise on aggregate production planning.