Controlling a Re-entrant Manufacturing Line via the Push–Pull Point

Dominique Perdaen, Dieter Armbruster, Karl G. Kempf and Erjen Lefeber

Abstract A reduced model of a re-entrant semiconductor factory exhibiting all the important features is simulated, applying a push dispatch policy at the beginning of the line and a pull dispatch policy at the end of the line. A commonly used dispatching policy that deals with short-term fluctuations in demand involves moving the transition point between both policies, the push–pull point (PPP) around. It is shown that with a mean demand starts policy, moving the PPP by itself does not improve the performance of the production line significantly over policies that use a pure push or a pure pull dispatch policy, or a CONWIP starts policy with pure pull dispatch policy. However, when the PPP control is coupled with a CONWIP starts policy, then for high demand with high variance, the improvement becomes approximately

---

Previously published in Perdaen, Dominique, Armbruster, Dieter, Kempf, Karl and Lefeber, Erjen (2008) 'Controlling a re-entrant manufacturing line via the push-pull point', International Journal of Production Research, 46 16:4521–4536. Reproduced here with permission.

D. Perdaen · E. Lefeber
Department of Mechanical Engineering,
Eindhoven University of Technology,
P.O. Box 513, 5600 MB, Eindhoven,
The Netherlands

E. Lefeber
e-mail: A.A.J.Lefeber@tue.nl

D. Armbruster (✉)
School of Mathematical and Statistical Sciences,
Arizona State University,
Tempe, AZ 85287-1804, USA
e-mail: armbruster@asu.edu

K. G. Kempf
Decision Engineering Group,
Intel Corporation, 5000 W. Chandler Boulevard,
Chandler, AZ 85226-3699, USA
e-mail: karl.g.kempf@intel.com

D. Armbruster and K. G. Kempf (eds.), *Decision Policies for Production Networks*,
DOI: 10.1007/978-0-85729-644-3_5,

a factor of 4. The unexpected success of a PPP policy with CONWIP is explained using concepts from fluid dynamics that predict that this policy will not work for perishable demand. The prediction is verified through additional simulations.

Keywords Re-entrant production · CONWIP · Dispatch policy

1 Introduction

A very important feature of the production of semiconductor wafers is the re-entrant line: Wafers are produced in layers and hence after one layer is finished a wafer returns to the same set of machines for processing of the next layer. Modern semiconductors may have on the order of 20–30 such layers. It is typical for wafers to spend several weeks in such a re-entrant production line, much of the time waiting for available machines. Process control in such long production lines with thousands of wafer and hundreds of processing steps making tens of different products is a special challenge. Most of the time the demand fluctuates on a much faster timescale than the factory cycle time, making it very difficult to use starts policies to react to the demand fluctuations. Typically, for a product with a constant mean demand, the mean demand is started. Due to stochasticity in the production and due to variation in the demand there is nevertheless a large mismatch in daily outputs and demand. In practice, to reduce the mismatch, production targets over a certain time horizon are given and wafers at the end of the production process are sped up or slowed down using dispatch policies. We are not concerned here with longer and larger fluctuations that might require an adjustment of the starting rate to cover changes of the desired WIP level as discussed in [14].

The combination of lot release and dispatching strategies is called Workload (or Flow) Control. An overview of state-of-the-art published research on workload control as applied to semiconductor industry is provided in [7]. A thorough overview of the literature on order release as a flow control is provided in [4], whereas [12] and [5] are two thorough surveys of the dispatching literature. Commonly used dispatching policies include: First-In, First-Out (FIFO), Earliest Due Date (EDD), Weighted Shortest Processing Time (WSPT), Least Slack (LS) and Least Setup Cost (LSC). In the seminal paper [16] many of these lot sequencing rules as well as a variety of input controls have been evaluated using simulation models of representative but fictitious semiconductor fabs. The main conclusion was that order release is more important than dispatching (30–40% change versus less than 10%), though there is an important connection between these decisions. Dynamic scheduling studies were done by [3] who implemented learning of dispatch rules in their simulation environment. Pure push and pull dispatch policies were studied by [2].

Most of the time demand fluctuates on a much faster timescale than the factory cycle time. Unfortunately, almost no literature exists on how to deal with the impact of a production surge or short-term increase in wafer starts that occurs when unexpected orders are received by a fab that is operating close to its designed capacity. In [6, 9, 11] some preliminary investigations into the surge problem have been done.

In order to deal with these short-term variations in demand we consider a dispatching policy which to the authors' knowledge has not been considered in the literature before, but which is used in practice. We simulate a reduced model of a re-entrant semiconductor factory exhibiting all the important features, applying a push (dispatch) policy at the beginning of the line and a pull (dispatch) policy at the end of the line. Here a push (pull) policy refers to the fact that a machine that is able to process more than one step gives priority to the earlier (later) step. Push policies are also known as first-buffer-first-served and pull policies are known as shortest-expected-remaining-process-time policies. We use a push policy upstream and a pull policy downstream. The step at which we switch from a push to a pull policy is called the push–pull point (PPP). Its dynamics is the control variable. Our objective (metric) is to reduce the mismatch between daily outputs and demand over a long time interval. We assume that over that time interval the demand has a constant mean demand and varies stochastically around the mean. By focussing on the output, this study complements the important work by [10] who were not concerned with output but with the behavior of the mean and variance of the cycle times as a function of different scheduling policies.

We show that with a policy that starts the mean demand, moving the PPP by itself does not improve the performance of the production line significantly over a pure push, a pure pull policy or a pure CONWIP starts policy [13] with pure pull dispatch. However, when the PPP dispatch control is coupled with a CONWIP starts policy, then for high demand with high variance, the improvement becomes approximately a factor of 4. We explain the unexpected success of a PPP policy with CONWIP using concepts from fluid dynamics that predict that this policy will not work for perishable demand. We verify this prediction.

2 The Factory Model

Our basic factory model consists of 26 production steps executed on nine machine sets. Table 1 contains all the specifications of this model. The first six machines are called diff1, diff2, litho1, etch clean, etch1 and ion impl, corresponding to production steps associated with diffusion, photolithography, etching and ion implantation respectively. They are associated with the transistor section of the production line and a wafer performs four loops through these machines in a specific order as indicated in Table 1. The last three machine sets are called metal dep, litho2 and etch2, corresponding to production steps that generate metal layers for interconnection of the transistors. The wafer loops through the metalization section of the production line twice. The transistor and metal loops are completely disjoint and do not share equipment. Rows 1–26 in Table 1 correspond to the 26 production steps. The entries in each row indicates the machine set that performs the step and the processing time spent in a machine in the set. For instance, step 3, 6, 10 and 14 are all performed on the photolithography machine litho1 with cycle times of 1, 1.25, 1 and 1.25 h, respectively.

Table 1 Factory model

	Diff 1	Diff 2	Litho 1	Etch clean	Etch 1	Ion impl	Metal dep	Litho 2	Etch 2	
Step	0	1	2	3	4	5	6	7	8	Station #
1				0.25						Clean wafer
2	8.00									Grow a layer
3			1.00							Pattern it
4					1.00					Etch away some
5		6.00								Grow a layer
6			1.25							Pattern it
7						2.50				Implant ions
8				0.50						Remove mask
9	7.00									Grow a layer
10			1.00							Pattern it
11					1.00					Etch some away
12				0.25						Clean wafer
13		5.00								Grow a layer
14			1.25							Pattern it
15						3.50				Implant ions
16				0.50						Remove mask
17								1.50		Pattern contact
18									1.75	Etch contact
19							2.25			Layer metal
20								1.00		Pattern metal
21									2.25	Etch metal
22								1.50		Pattern contact
23									2.00	Etch contact
24							2.25			Layer metal
25								1.00		Pattern metal
26									2.50	Etch metal
	15.00	11.00	4.50	1.50	2.00	6.00	4.50	5.00	8.50	Total hours required per lot
	750	550	900	300	400	1200	900	1000	1700	Total hours needed per week
	0.80	0.75	0.90	0.60	0.75	0.85	0.85	0.90	0.55	Average availability
	134.40	126.00	151.20	100.80	126.00	142.80	142.80	151.20	92.40	Total hours avail per machine
	5.58	4.37	5.95	2.98	3.17	8.40	6.30	6.61	18.40	Minimum num. tools needed
	1.25	1.25	1.00	1.25	1.50	1.10	1.25	1.05	1.10	Constraint degree desired
	6.98	5.46	5.95	3.72	4.76	9.24	7.88	6.94	20.24	Number of tools needed
	7	6	6	4	5	10	8	7	21	Number of tools installed

The second part of Table 1 is a spreadsheet calculation to determine the required number of machines (tools) to have a production target of 200 lots per week, given availability rates of the machines and desired levels of contraints for a given machine set. Consider for instance the last 8 rows in the column litho1: A wafer spends a total of 4.5 h in litho1. Hence to produce 200 wafers per week we need 900 h per week of machine time. Assuming that a litho1 machine is 90% available and a work week of 168 h this machine works for 151.2 h per week and hence we need 5.95 machines of that type. Since this is a very expensive machine, it is planned to be the bottleneck and hence has a constraint factor of 1.0. As a result six machines will be installed. Taking into account that the diffusion machines batch four wafers per machine cycle we reach the installation targets in the last row in a similar way for all columns.

This model is implemented as a discrete event simulation in χ [15, 8] a specification language developed at the Eindhoven University of Technology. Stochasticity enters the simulation at various levels: The time that a machine is in service, and the time that it is not, is distributed by a Weibull-distribution [8] with a mean "in service" time of 10 process times and a variance of 50%. The demand is randomly generated and is fixed for a simulation.

The actual processing times are pulled out of an Exponential-distribution [8] with the mean equal to the process times in Table 1. Note that, while the raw processing times of semiconductor processing machines are narrowly distributed, the unloading of machines depends on the availability of human operators and is highly variable. Nevertheless using an exponential distribution probably constitutes a worst case scenario for a practical model. Overall the stochastic parameters are fixed in a way, such that simulations of the model generate an outflux variance of 20% around the nominal influx of 200 per week, i.e. the throughput varies between 160 and 240 wafers per week.

3 The Push–Pull Point Algorithm

The goal of the PPP policy is to reduce the mismatch between fluctuating demands and the stochastically varying outflux of the factory. This policy divides the production line in two parts. Upstream of the PPP, priorities are assigned using a push strategy, downstream they are assigned according to a pull strategy. In conflicts across the PPP we always give priority to the steps in the pull-part. Figure 1 shows a typical priority assignment.

The PPP is moved depending on the demand: Given a demand period and a distribution of the work in progress (WIP) over the queues of all production steps (the WIP-profile), we place the PPP at such a point that the WIP downstream from the PPP is equal to the demand in the chosen demand period. When the demand increases, more products have to be pulled out of the line moving the PPP upstream. When the demand decreases, the PPP will shift downstream.

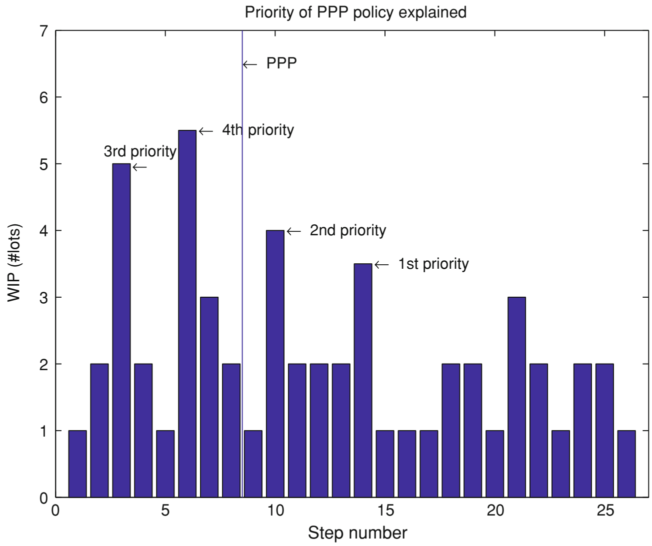

Fig. 1 The priority distribution when the PPP-policy is used

The possible success of such a strategy is based on three important facts:

- The clearing function [1], i.e. the throughput as a function of the load in the factory in steady state is significantly higher for a production line run completely with a push dispatch policy than for one run completely with a pull dispatch policy. Hence by increasing or decreasing the part of the production line that is run in pull policy we temporarily should increase or decrease the outflux. We show below the details of this effect for our model production line.
- The location of the push–pull point determines the average shape of the WIP profile in steady state. In particular, on average WIP decreases in the queues downstream of the PPP and increases upstream from the PPP. Figure 1 shows this schematically for the queues in front of the photolithography machines for a fixed PPP point. Figure 2 shows that this is true to a large extent for simulations on average, even when the PPP point is dynamically moved.
- The cycle time through the factory and the time between readjustments of the PPP have to be related. In particular, if adjusting the PPP according to demand on average places the PPP approximately in the middle of the production line adjusting to higher and lower demand by changing the PPP should be feasible.

4 Results

To determine the effectiveness of the PPP strategy we compare it to simulations with a starts policy of the mean demand and dispatch policies of pure push, pure pull as well as a CONWIP starts strategy using a dispatch policy of pure pull. We have

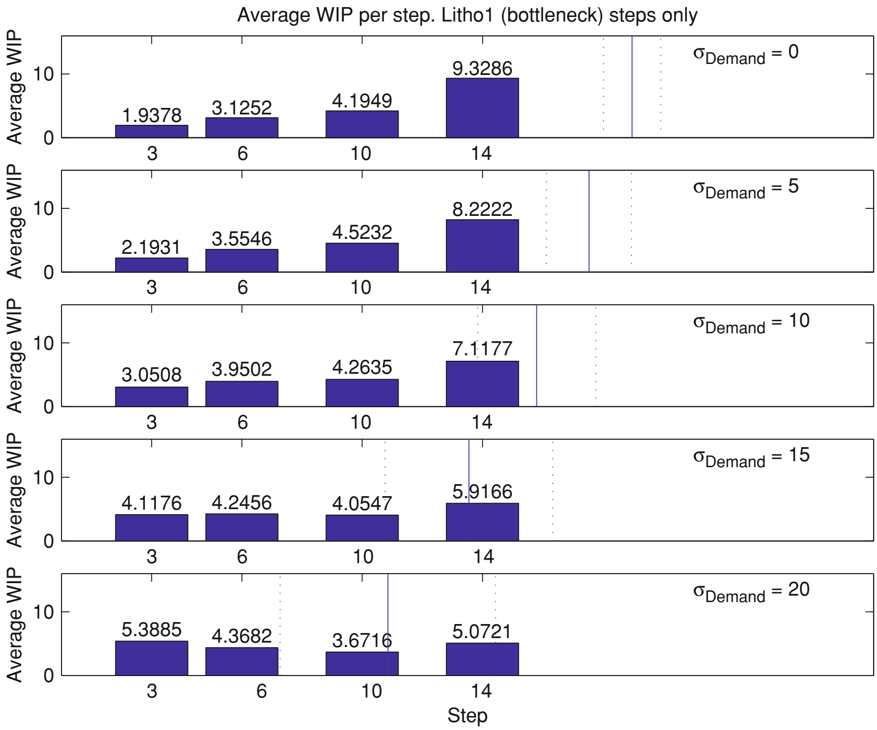

Fig. 2 Average queue length at the litho1 steps. The vertical line and its two dashed sidebars are the average position of the PPP plus/minus 2σ. As the PPP point moves upstream the WIP in the last two photolithography steps decreases and the WIP in the first two photolithography steps increases

also combined the PPP strategy with CONWIP as a starts policy. In all simulations we employ a FIFO policy within a given queue for a given production step. We run 500 simulations per data point. The demand $d(t)$ for each simulation is generated independently by choosing a demand for a two day period out of a normal distribution (throwing away the rare events that gave negative demands) with an average of 180 lots per week. The demand is not perishable, which means that the backlog or the inventory of the previous demand period is taken into account for the present demand period. The PPP is adjusted every 2 days (one demand period). Since the cycle time for our simulation factory is in the order of 5 days, the two day readjustment time places the PPP well inside the production line. The simulation-time for every single run is 144 weeks. The different control strategies are compared using the absolute value of the mismatch between output and demand over each demand period. Mismatch $m(t)$ and costs are given as

$$m(0) = 0 \tag{1}$$

$$m(i) = m(i-1) + d(i) - o(i) \tag{2}$$

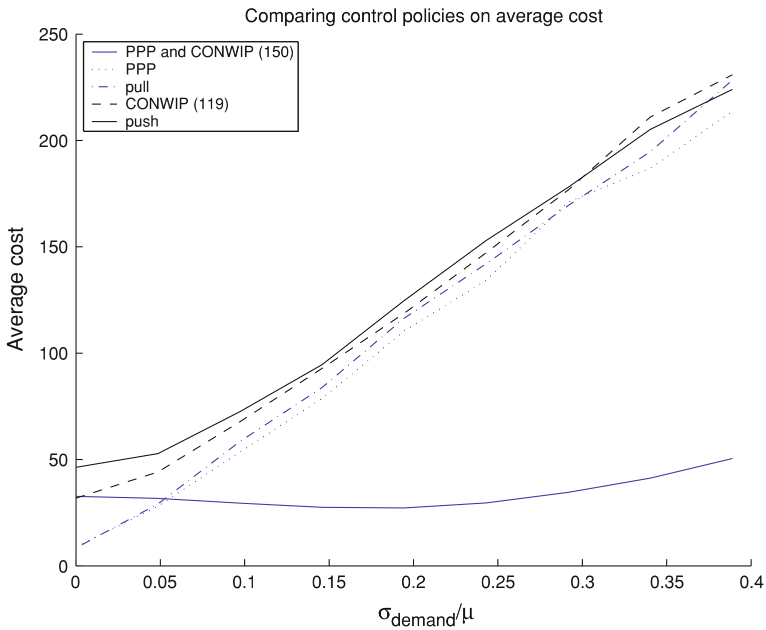

Fig. 3 Average costs per simulation for different control strategies as a function of the coefficient of variation of the demand

Table 2 Variance of cost as a function of the variation of the demand

$\sigma_{\text{demand}}/\mu$	0	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4
σ^2_{cost}	3.0	3.7	6.6	7.9	9.7	27.8	75.1	216.6	578.1

$$\text{cost}(t) = \sum_{i}^{t} |m(i)|. \tag{3}$$

Here $o(t)$ is the output of the factory plus backlog and storage, i.e. over and under-production cost the same 1$ per lot per demand interval (2 days).

Figure 3 shows the average costs over 500 simulations as a function of the variance in the demand for all the different strategies. Table 2 shows the variances for the nine simulation points in Fig. 3.

The results are surprising: Pure push, pure pull, regular PPP (all with mean demand starts policy) and a CONWIP starts policy (pure pull dispatch policy) with a WIP level of 119 lots all increase monotonically with the demand variation and have very similar average cost. In contrast to that, a policy that combines the starts policy of a CONWIP rule and a WIP of 150 lots with the PPP control policy has almost constant costs over a wide range of demand variations. In addition the costs for high

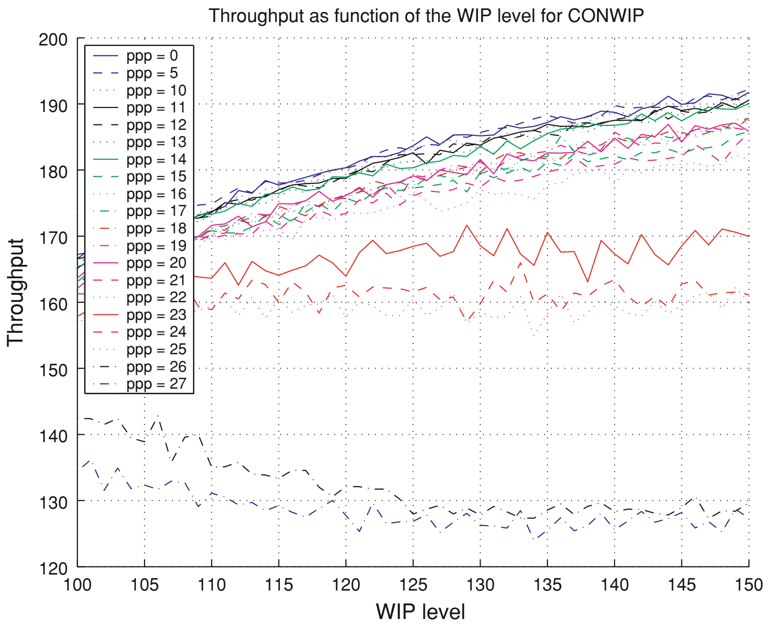

Fig. 4 Throughput as a function of total WIP for CONWIP policies with fixed push–pull points

demand variations are significantly lower for the PPP with CONWIP than for the other policies—50$ versus more than 200$.

5 Analysis of the PPP-CONWIP Policy

Figure 4 begins to explain the success of the PPP-CONWIP policies. It shows the clearing functions for CONWIP policies with different fixed push–pull points. The curve indicated with $ppp = 0$, corresponding to a pure pull dispatch policy, gives the highest throughput of all possible policies. The curve labeled $ppp = 27$ is a pure push dispatch policy that gives the lowest throughput of all. The intermediate curves indicated by $ppp = x$ denote a dispatch policy where the push–pull point has been fixed at step x. Note that for a complete push policy the throughput actually decreases with an increase in WIP. This is the result of an interplay between the back loaded WIP distribution of the push policy and the batching in the diffusion steps. Figure 4 also explains the choice of a CONWIP starts policy with a WIP level of 119 lots for a pure pull dispatch policy used in Fig. 3: The top curve in Fig. 4 represents a pure pull dispatch policy. The associated WIP level in steady state for a throughput of 180 lots/week is 119 lots which we use as the desired WIP level [14].

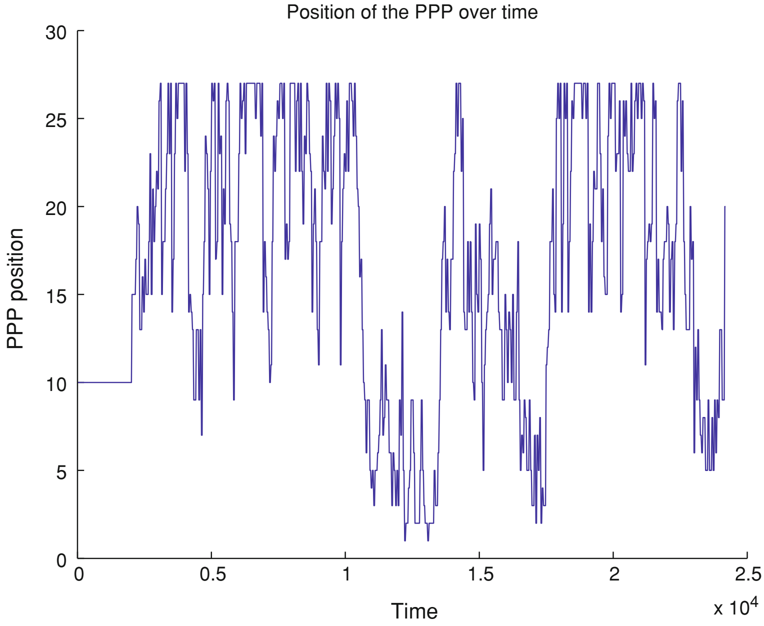

Fig. 5 Time evolution of the push–pull point as a function of time for a PPP-CONWIP policy with WIP level 130

These clearing functions suggests one reason for the success of the PPP-CONWIP policy: By using a CONWIP starts policy with a high WIP level and switching the PPP, we can change the outflux in the factory by a significant amount. For instance, for the WIP level of 150 lots we can get throughputs between approximately 130 and 190 per week. Note also that there is no good push–pull point for a WIP level of 150 that creates the throughput of 180 per week that we are using for our simulations. A PPP at stage 1–15 creates a throughput much higher and a PPP at stage 15–26 creates a throughput much lower than 180 per week. As a result, a completely deterministic demand cannot use a fixed PPP even though the demand is constant and hence has to jump back and forth, creating extra backlog or overproduction cost. This is the reason for the slight increase in cost for the PPP-CONWIP policy with WIP level 150 in Fig. 3 for low demand variation.

A different issue explains the failure of the pure PPP dispatch policy to be much better than a regular pull dispatch policy. Assume a push–pull point in the middle of the production line and an increase in demand. In response we will move the PPP upstream and clear out more of the WIP than we usually do over the demand period. However, we will only *start the average* amount. Consequently, WIP goes down and a second increase in demand will move the PPP rapidly further upstream. As a result we easily reach the point where the PPP is at the beginning of the line and the policy becomes a pure push dispatch policy. We cannot further increase the outflux than that. Similarly, a demand signal that has several periods below average will eventually move the PPP to the end of the factory and hence constitute a pull policy. We cannot reduce the outflow further than that. A CONWIP starts policy reduces the instances that the push–pull point is at one of the extremes of the production line by instantaneously starting more when more was pulled out of the factory and starting less if more was left in the factory. Figures 5 and 6 show the position of the PPP as a function of time for a PPP-CONWIP and a free PPP policy, respectively. Clearly the free PPP policy gets locked into pure push or pure pull policies much more often than the PPP-CONWIP.

We can illustrate the difference between free PPP and PPP-CONWIP policies with the following illustration based on fluid flows. For the purpose of this illustration let us consider the average behavior of a large number of lots as they move through the factory. We assume that the average speed $v(t)$ of a lot for a factory that is in steady state is constant over all production steps and depends on the dispatch policy. In particular, the average cycle time for a lot under a pull (dispatch) policy is shorter than for a lot produced under a push (dispatch) policy. Hence the associated average velocity for a pull policy is higher than that for a push policy. Let us consider a continuum of production steps and a continuum of lots such that we can define a WIP density $\rho(x, t)$ that describes the density of lots at stage x at time t. Then the throughput of the factory becomes $\lambda(x, t) = \rho(x, t)v$. In steady state, the throughput is constant and hence we get a constant WIP profile $\rho(x) = \frac{\lambda}{v}$ that does not depend on t because we are looking at steady state and does not depend on x, because we assume v to be constant. This is certainly not exactly true but a good approximation for the purpose of this illustration. Now, for a PPP policy we can consider the upstream part of the production line as a homogeneous push line and the downstream part as a homogeneous pull line, each with its own constant velocity with $v_{\text{push}} < v_{\text{pull}}$. Since the throughput is the same everywhere and since $\rho v = \lambda$ has to hold, we get a jump in the WIP profile at the push–pull point by the amount

$$\frac{\rho_{\text{push}}}{\rho_{\text{pull}}} = \frac{v_{\text{pull}}}{v_{\text{push}}}. \tag{4}$$

Figure 7a shows the constant throughput and the discontinuous WIP profile.

Assume we now move the PPP upstream by an amount Δx instantaneously. The queues that were just upstream of the PPP and hence had the lowest priority on the line now move up in priority and therefore speed up. In other words, part of the WIP profile that used to be in the push region and had a high WIP level now is in the pull region. As the velocity in the pull region is higher, the product of $\rho_{\text{push}} v_{\text{pull}} > \lambda$, i.e. we create a flux bump. Similarly we create a flux dip by moving the PPP downstream. The flux changes are

$$q \cdot \Delta x = \lambda \frac{v_{\text{pull}}}{v_{\text{push}}}, \tag{5}$$

$$q \cdot \Delta x = \lambda \frac{v_{\text{push}}}{v_{\text{pull}}}, \tag{6}$$

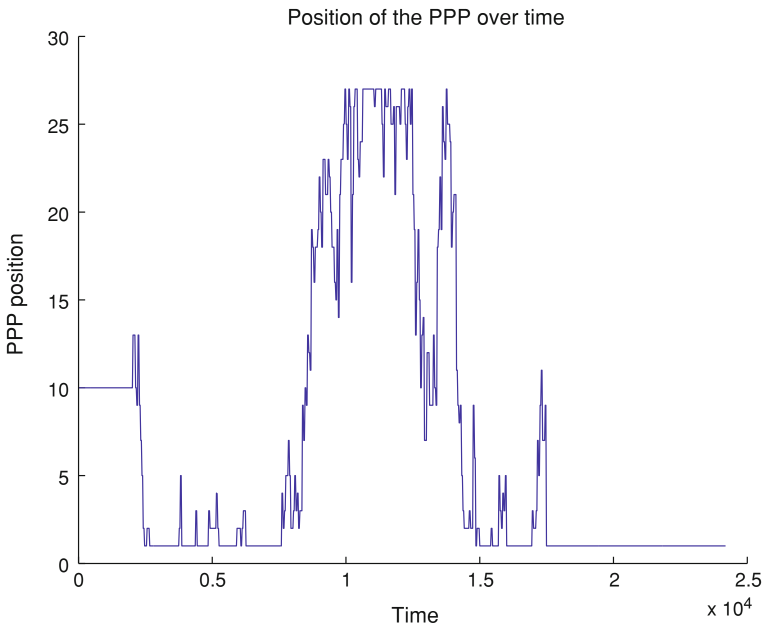

Fig. 6 Time evolution of the push–pull point as a function of time for a free PPP policy

for the flux bump and flux dip, respectively. Keeping the PPP at its new location the flux bump is downstream from the PPP and hence moves downstream with the constant speed v_{pull} pulling a WIP bump with it until they both exit the factory. During the time they exit they will increase the outflux. Depending on the remaining processing time from the push–pull point to the end of the production line, the increase in outflux may or may not happen within the demand time interval. Figure 7b and c show this time evolution. After the WIP/flux bump has exited, the total WIP in the factory is lower and hence in order to satisfy the same demand, the push pull point will have to move yet further upstream driving it toward the beginning of the factory.

In contrast, the time evolution of the flux bump for the PPP-CONWIP policy is illustrated in Fig. 8.

As the CONWIP starts policy is implemented by matching the starts to the outflux, once the WIP bump moves out of the factory, the starts will be increased to create a new WIP bump. In that way, the total throughput will stay high until the PPP point is moved downstream again. That will happen when the backlog has moved to zero and the sum of actual backlog and actual demand has decreased. In that way we have a policy that reverts all the time to a match between demand and outflux. This explanation can be checked by running the simulation with a perishable demand protocol: We only register whether there is a mismatch of the current outflux and the current demand but do not try to make up for that mismatch on the next time interval. For such a model the PPP-CONWIP policy should not be better than the

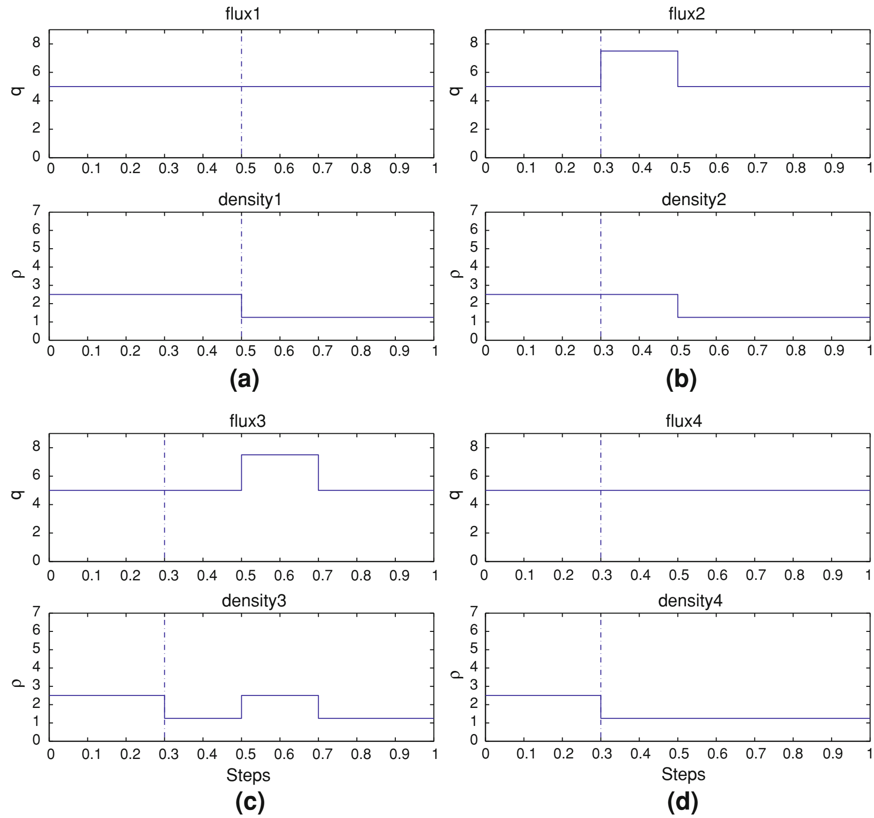

Fig. 7 Stages of creating a flux-bump

free PPP policy. The only thing that matters is whether the flux bump or flux dip that is created arrives at the end of the factory within the demand time window. Our simulations confirm this: PPP and PPP-CONWIP policies behave very similarly and do not improve the performance of the production line appreciably with perishable demand.

6 Conclusion

We have studied process control in a reduced model of a re-entrant semiconductor factory using discrete event simulations. We showed that when running a factory with a push dispatch policy at the beginning of the factory and a pull dispatch policy at the end of the factory while using an average demand starts policy, the transition

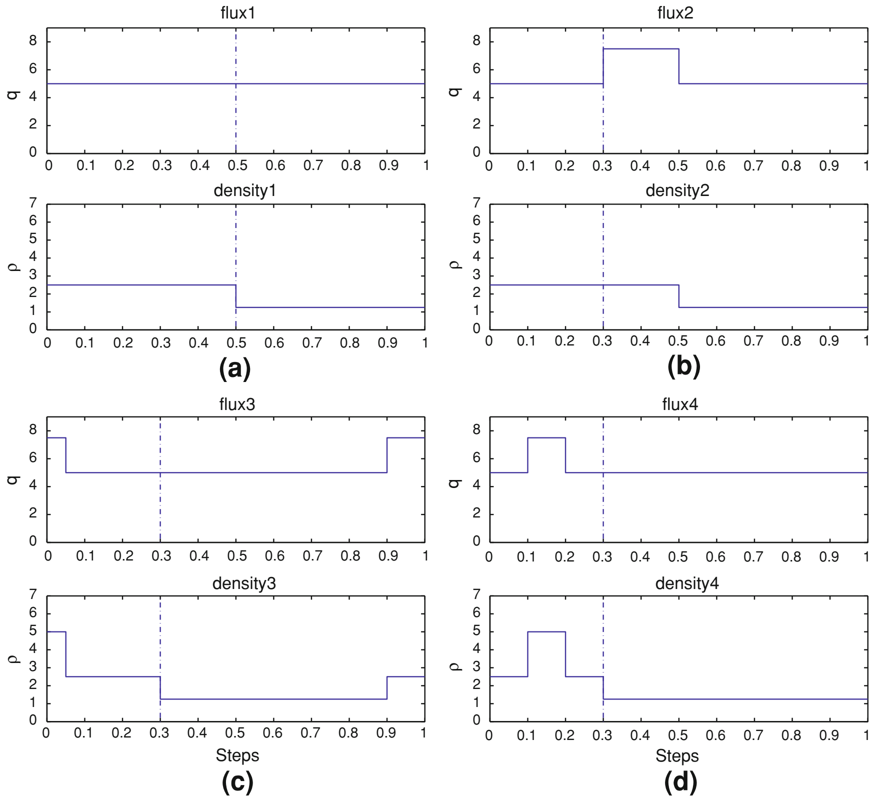

Fig. 8 Stages of creating a flux-bump for a PPP-CONWIP policy

point (the PPP) can be used to reduce the mismatch between stochastic outfluxes of the factory and stochastic demands.

We have two results that are of immediate practical interest:

1. A pure PPP dispatch policy that reaches into the factory from the end and pulls out the desired demand will not significantly reduce the mismatch between outflux and demand for a demand signal that has a constant average and varies stochastically around that average.
2. A PPP dispatch policy coupled with a CONWIP starts policy adjusted for a WIP level that allows maximal flux changes through moving the PPP will significantly reduce the mismatch for a production with non-perishable demand.

Process control in these re-entrant production lines is very difficult since only starts policies and dispatch rules are the obvious control actuators that influence the outflux of the factory. However, as a byproduct of this study we have identified another control parameter: The actual WIP profile will be very important for the success of a PPP policy. It seems likely that very homogeneous WIP profiles are better for the control action of the PPP policy than the WIP profile that we have currently examined. Those WIP profiles are determined by the level of constraint we are choosing for a particular machine set. It will be an interesting further study to determine the interplay of the constraint levels and the PPP policy.

Acknowledgments The research of D.A. was supported by NSF grants DMS-0604986 and DMS-0204543. We thank Ton Geubbels for help in developing the χ—model.

References

1. Asmundsson J, Rardin RL, Uzsoy R (2003) Tractable non-linear capacity models for aggregate production planning. Technical Report IN 47907–1287, Purdue University, School of Industrial Engineering, West Lafayette
2. Atherton RW, Dayhoff JE (1986) Signature analysis: Simulation of invetory, cycle time and throughput tradeoffs in wafer fabrication. IEEE Trans Compon Hybrids Manuf Technol 9(4):498–507
3. Aytug H, Koehler GJ, Snowdon JL (1994) Genetic learning of dynamic scheduling within a simulation environment. Comput Oper Res 21(8):909–925
4. Bergamaschi D, Cigolini R, Perona M, Portioli A (1997) Order review and release strategies in a job shop environment: A review and classification. Int J Prod Res 35(2):399–420
5. Blackstone JH, Phillips DT, Hogg GL (1982) A state-of-the-art survey of dispatching rules for job shop operations. Int J Prod Res 20(1):27–45
6. Dummler MA (2000) Analysis of the instationary behavior of a wafer fab during product mix changes. In: Proceedings of the winter simulation conference, pp 1436–1442
7. Fowler JW, Hogg GL, Mason SJ (2002) Workload control in the semiconductor industry. Prod Plan Control 13(7):568–578
8. Hofkamp AT, Rooda JE (2002) χ Reference manual. Eindhoven University of Technology, Systems Engineering Group, Department of Mechanical Engineering. http://se.wtb.tue.nl/documentation
9. Kato K (1996) Lot start evaluation system using simulator. In Proceedings of international symposium on semiconductor manufacturing, Tokyo, Japan, pp 293–296
10. Lu S, Ramaswamy D, Kumar PR (1994) Efficient scheduling policies to reduce mean and variance of cycle time in semiconductor plants. IEEE Trans Semicond Manuf 7:374–388
11. McKiddie R (1995) Some ‘no-panic’ help for wafer-start surges. Semicond Int, 115–120
12. Panwalkar SS, Iskander W (1977) Capacity planning and control. Oper Res 25(1):45–61
13. Spearman ML, Woodruff DL, Hopp WJ (1994) CONWIP: A pull alternative to kanban. Int J Prod Res 28(5):879–894
14. Sterman JD (2000) Business dynamics, systems thinking and modeling for a complex world. McGraw-Hill, New York
15. Vervoort J, Rooda JE (2003) Learning χ. Technical report, Eindhoven University of Technology, Systems Engineering Group, Department of Mechanical Engineering. http://se.wtb.tue.nl/documentation
16. Wein LM (1988) Scheduling semiconductor wafer fabrication. IEEE Trans Semicond Manuf 1(3):115–129