

České vysoké učení technické v Praze

Fakulta elektrotechnická

Katedra elektrotechnologie



Bakalářská práce

Využití simulačního nástroje AnyLogic při simulaci procesů.

Autor: Igor Sarsembayev

Studijní obor: Aplikovaná elektrotechnika

Vedoucí bakalářské práce: Ing. Martin Molhanec, CSc.

Praha, 2018

Zadání bakalářské práce

Student: Igor Sarsembayev
Fakulta: Fakulta elektrotechnická
Katedra: Katedra elektrotechnologie
Studijní program: Elektrotechnika, energetika a management
Studijní obor: Aplikovaná elektrotechnika
Název bakalářské práce: Využití simulačního nástroje AnyLogic při simulaci procesů

Pokyny pro vypracování:

1. Vyberte vhodný praktický proces k simulaci.
2. Pro simulaci využijte simulační nástroj AnyLogic.
3. Sestrojte simulační model v tomto nástroji, otestujte ho na správnost a proveďte vhodné simulační experimenty.
4. Vytvořený model podrobně dokumentujte.
5. Zhodnoťte dosažené výsledky jednak z hlediska vhodnosti využití nástroje AnyLogic a jednak z hlediska dosažených výsledků.

Seznam doporučené literatury:

1. Manuály k programu AnyLogic.
2. Modelování a simulace komplexních systémů – Radek Pelánek.

Vedoucí: Ing. Martin Molhanec, CSc.

Platnost zadání: 30.09.2019

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Sarsembayev** Jméno: **Igor** Osobní číslo: **452867**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra elektrotechnologie**
Studijní program: **Elektrotechnika, energetika a management**
Studijní obor: **Aplikovaná elektrotechnika**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Využití simulačního nástroje AnyLogic při simulaci procesů.

Název bakalářské práce anglicky:

Using of AnyLogic tool for process simulation.

Pokyny pro vypracování:

1. Vyberte vhodný praktický proces k simulaci.
2. Pro simulaci využijte simulační nástroj AnyLogic.
3. Sestrojte simulační model v tomto nástroji, otestujte ho na správnost a proveďte vhodné simulační experimenty.
4. Vytvořený model podrobně dokumentuje.
5. Zhodnoťte dosažené výsledky jednak z hlediska vhodnosti využití nástroje AnyLogic a jednak z hlediska dosažených výsledků.

Seznam doporučené literatury:

1. Manuály k programu AnyLogic
2. Modelování a simulace komplexních systémů - Radek Pelánek

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Martin Molhanec, CSc., katedra elektrotechnologie FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **19.02.2018** Termín odevzdání bakalářské práce: **08.01.2019**

Platnost zadání bakalářské práce: **30.09.2019**

Ing. Martin Molhanec, CSc.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Ing. Pavel Ripka, CSc.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studenta

Čestné prohlášení

Prohlašuji, že jsem celou zadanou bakalářskou práci „Využití simulačního nástroje AnyLogic při simulaci procesů“ včetně vytvořených simulačních modelů zpracoval samostatně.

V Praze dne

Igor Sarsembayev

Anotace (CZ)

Tato bakalářská práce vypraví o simulačním nástroji AnyLogic určeném k sestrojení simulačních modelů různých typů modelování. Cílem je seznámit profesori a studenty s možnostmi simulačního modelování. Jako příklad k praktickému zpracování modelu byla zvolena simulace výroby polovodičových systémů pro mikroprocesory. K sestrojenému simulačnímu modelu se vytváří vhodný experiment pro posouzení výsledků a zvolení nejlepší varianty provedení.

Klíčová slova

Křemíkové desky, agent, blok, element, AnyLogic, PLE, model, modelování, simulace, proces, experiment, parametr, proměnná.

Annotation (EN)

This Bachelor Thesis deals with the AnyLogic simulation tool designed to construct simulation models of various types of modeling. The purpose is to familiarize professors and students with the possibilities of simulation modeling. As an example of practical modeling, a simulation of manufacturing semiconductor systems for microprocessors was chosen. A suitable experiment is made to assess the results and select the best version

Keywords

Silicon wafer, agent, block, element, AnyLogic, PLE, model, modeling, simulation, process, experiment, parameter, variable.

Obsah

| | |
|--|-----------|
| 1. Úvod | 9 |
| 2. Použité teorie a metody | 10 |
| 3. Současný stav problematiky, modelování | 12 |
| 4. Návrh řešení – AnyLogic | 14 |
| 5. Popis procesu | 21 |
| 5.1 „Clean-room“ | 21 |
| 5.2 Výroba krystalického ingotu a křemíkových destiček..... | 21 |
| 5.3 Design | 22 |
| 5.4 Diffusion area (oblast difúze) | 22 |
| 5.5 Photolithography area (oblast fotolitografie)..... | 22 |
| 5.6 Ion implantation area (oblast implantace iontů) | 24 |
| 5.7 Chemical vapor deposition area (oblast chemické depozice z plynné fáze)..... | 24 |
| 5.8 Etching area (oblast leptání) | 25 |
| 5.9 Physical vapor deposition area (oblast fyzikální depozice z plynné fáze) | 25 |
| 5.10 Chemical mechanical polishing area (oblast chemicko-mechanického leštění).... | 25 |
| 6. Simulační model továrny | 26 |
| 6.1. Model vstupu pracovníků do čistého výrobního prostoru | 26 |
| 6.2. Model procesu výroby polovodičů pro mikroprocesory..... | 29 |
| 7. Experiment | 35 |
| 8. Závěr | 37 |

Seznam funkcí, kódů, vzorců a tabulek

| | |
|------------------------------|----|
| Funkce 1, triangularAV | 11 |
| Kód 1..... | 31 |
| Kód 2..... | 33 |
| Kód 3..... | 34 |
| Kód 4..... | 35 |
| Kód 5..... | 35 |
| Vzorec 1 | 21 |
| Vzorec 2..... | 21 |
| Vzorci 3, 4, 5, 6 | 22 |
| Tab. 1 | 24 |

Seznam obrázků

| | |
|--|----|
| Obr. 1, Grafické zobrazení funkce exponential..... | 10 |
| Obr. 2, Popis funkce exponential | 10 |
| Obr. 3, Popis funkce triangular..... | 10 |
| Obr. 4, Grafické zobrazení funkce triangular | 10 |
| Obr. 5, Grafické zobrazení funkce triangularAV | 11 |
| Obr. 6, Grafické zobrazení funkce uniform..... | 11 |
| Obr. 7, Popis funkce uniform | 11 |
| Obr. 8, Rozhraní AnyLogic | 14 |
| Obr. 9, Nástroje AnyLogic | 16 |
| Obr. 10, Blok "source" | 17 |
| Obr. 11, Blok "delay" | 17 |
| Obr. 12, Blok "queue" | 17 |
| Obr. 13, Blok "sink" | 17 |
| Obr. 14, Element "stock" | 18 |
| Obr. 15, Element "flow" | 18 |
| Obr. 16, Element "table function" | 18 |
| Obr. 17, Ostatní elementy systémové dynamiky | 18 |
| Obr. 18, Element "statechart entry point" | 19 |
| Obr. 19, Element "state" | 19 |
| Obr. 20, Element "transition" | 19 |
| Obr. 21, Element "final state" | 19 |
| Obr. 22, Okno simulace | 20 |
| Obr. 23, Model vstupu do továrny | 26 |
| Obr. 24, Element "pedSource" | 26 |
| Obr. 25, Element "pedWait" | 27 |
| Obr. 26, Element "pedGoTo" | 27 |
| Obr. 27, Element "pedService" | 27 |
| Obr. 28, Fronty..... | 27 |
| Obr. 29, Element "pedGoTo1" | 27 |
| Obr. 30, Element "pedSink" | 28 |
| Obr. 31, Oblast modelu vstupu do továrny | 28 |
| Obr. 32, Blokové schéma jednoho z výrobních procesů | 31 |
| Obr. 33, Blokové schéma celého výrobního procesu | 32 |
| Obr. 34, Time measure start..... | 33 |
| Obr. 35, Time measure end..... | 33 |
| Obr. 36, Obsluha agentu | 34 |
| Obr. 37, Experiment ParametersVariation se zapnutou obsluhou | 42 |
| Obr. 38, Experiment ParametersVariation..... | 43 |

Seznam příloh

Bakalářská práce.rar – soubor obsahující sestavené modely

| | |
|--|----|
| Obr. 37, Experiment ParametersVariation se zapnutou obsluhou | 42 |
| Obr. 38, Experiment ParametersVariation..... | 43 |

Seznam použitých zkratek a symbolů

| | |
|-----------------|---|
| PLE | Personal learning edition |
| mm | milimetry |
| mm ² | milimetr čtverečný |
| nm | nanometry |
| cm ² | centimetr čtverečný |
| min | minuty |
| °C | stupeň Celsia |
| mJ | miliJouly |
| kV | kiloVolty |
| wph | wafer per hour – počet desek za hodinu |
| CVD | Chemical vapor deposition |
| PVD | Physical vapor deposition |
| CMP | Chemical mechanical polishing |
| mld | miliardy |
| Mb | megabyte |
| CMOS | Complementary Metal–Oxide–Semiconductor |
| RCA clean | standard set of wafer cleaning steps which need to be performed before high-temperature processing steps (oxidation, diffusion, CVD) of silicon wafers in semiconductor manufacturing |

1. Úvod

Cílem této bakalářské práce je seznámit studenty a profesory se simulačním prostředím programu AnyLogic (The AnyLogic Company), který mohl být vhodný ve více různých oblastech jak studia, tak i výzkumu. V průběhu napsání bakalářské práce byla využita verze programu „Personal Learning Edition“ (dále jen PLE), která je zadarmo ke stažení na příslušném webu. Tato verze má některá omezení a některé z funkcí jsou blokovány, nicméně AnyLogic PLE má dostatečně velký rozsah nástrojů, které umožňují jeho využití v rámci studia k simulování, podrobnějšímu a názornějšímu vysvětlení některých jevů anebo experimentům jak se změnou veličin vstupních, tak i průběžných, či veškerých jiných koeficientů a dat. Simulační prostředí AnyLogic bylo zvoleno kvůli jednoduchosti, lehce pochopitelnému rozhraní, které dovolí rychle naučení a postup k vlastní práci. Použití již mnohým známého programovacího jazyku Java, například pro zadání principu konání některých akcí prostřednictvím kódu, a možnost praktické ukázky zvoleného procesu pomocí animovaných objektů je výhodou. Zároveň podstatnou výhodou AnyLogic je možnost simulaci jakéhokoliv procesu, buď to demonstrace práci v kanceláři, anebo model fungování celé továrny. Pro tuto závěrečnou práci byla zvolena zjednodušená simulace výroby polovodičových systémů pro mikroprocesory.

Hlavní myšlenkou programu AnyLogic je poskytnout nástroj, který by mohl použít kdokoli pro své potřeby. Nástroj umožňuje provést důležité testy pro získání většinou číselných výsledků k posouzení, jak by mohly být ovlivněny procesy při změně některých parametrů, například kolik lidí by bylo potřeba pro výrobu určitého počtu zboží nebo pro vykonání práce v kanceláři, skladu a kdekoli jinde; jak zvětšit počet pracujících strojů, parametry strojů anebo dokonce i vyměnit stroje za jiné, aby v budoucnosti získali větší výnos a rozšířili výrobu. V rámci závěrečné práce budou vysvětlené základní principy funkcí simulačního prostředí, základní elementy instrumentáře, budou popsány jednotlivé bloky simulace a celý proces výroby polovodičů. Mým cílem je ukázat, že zvolené simulační prostředí je použitelné ve výuce a mohlo by zlepšit vysvětlení některých z procesů ve mnoha předmětech na vysoké škole.

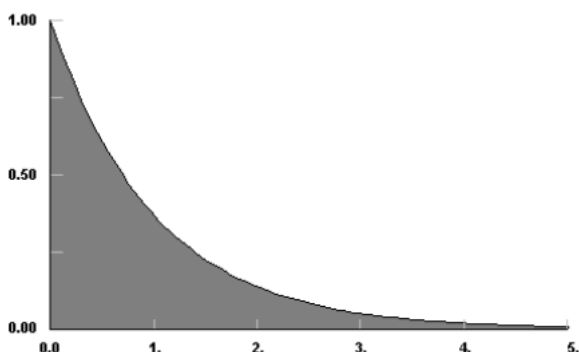
V následujících kapitolách bude vysvětlen současný stav problematiky modelování, návrh řešení pomocí simulačního prostředí programu AnyLogic, rozepsán postup výrobního procesu polovodičů od písku do mikroprocesoru, podrobně vysvětlen sestrojený model továrny, provedeny potřebné experimenty a na závěr prozkoumány výhody a nevýhody simulačního modelování a zejména AnyLogic, zhodnoceny výsledky provedené práce.

2. Použité teorie a metody

(1) Funkce rozdělení pravděpodobnosti použité pro zadání potřebných veličin elementů:

- `exponential(double λ , double min)`, kde λ je parametrem tvaru, a *min* je parametrem minimální hodnoty x .

Funkce `exponential()` se často používá pro zobrazení času mezi náhodnými událostmi, jako čas mezi vstupy v označené oblasti v modelu fronty nebo čas mezi havárií v modelu zkoušení spolehlivosti.



$$f(x; \lambda) = \begin{cases} \lambda e^{-\lambda x} & , x \geq 0, \\ 0 & , x < 0. \end{cases}$$

Obr. 2, Popis funkci exponential

Obr. 1, Grafické zobrazení funkci exponential

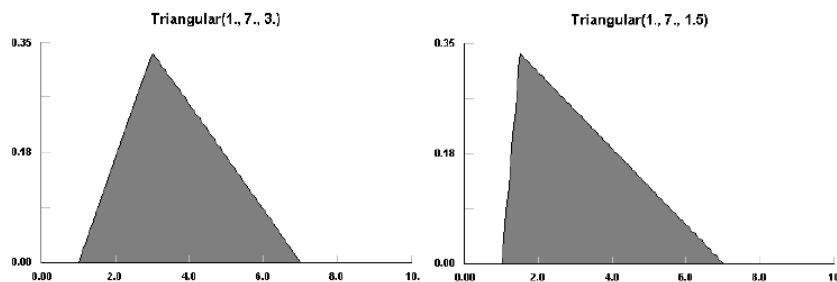
- `triangular(double min, double max, double mode)`, kde *min* je minimalni hodnota, *max* je hodnota maximalni a *mode* je veličinou s největší pravděpodobnosti v probíranem procesu.

Tato funkce se používá, kdy je velmi málo dat anebo data nejsou vůbec. Tato funkce není příliš přesná, proto není vhodné je použít pro popis důležitého procesu, nicméně může to být využito pro docela náhodné procesy, ve kterých není potřeba mít velkou přesnost.

$$f(x) = \begin{cases} \frac{2(x - \min)}{(\max - \min)(\text{mode} - \min)} & \min < x \leq \text{mode} \\ \frac{2(\max - x)}{(\max - \min)(\max - \text{mode})} & \text{mode} < x \leq \max \end{cases}$$

min = minimum x
max = maximum x
mode = most likely

Obr. 3, Popis funkci triangular



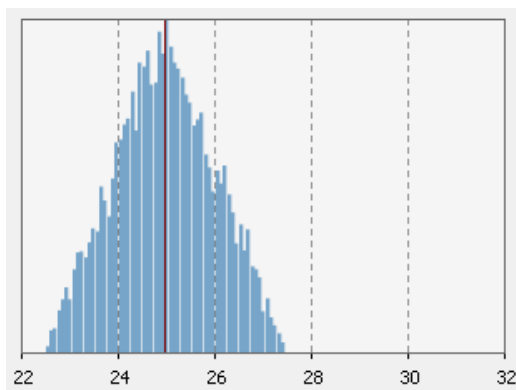
Obr. 4, Grafické zobrazení funkci triangular

- `triangularAV(double average, double variability)` – funkce se používá pro distribuci parametrů v zadaném rozmezí, například: „požadovaná hodnota +/- 20%“.

Má stejný smysl jako funkce `triangular` zapsaná následujícím způsobem:

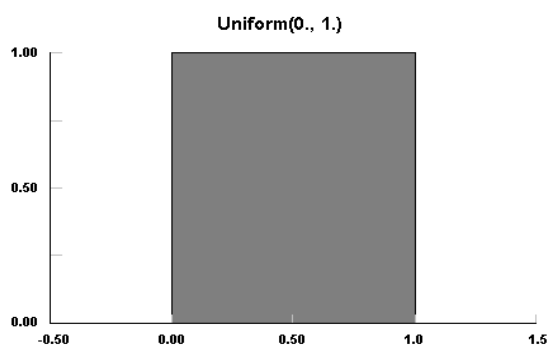
`triangular(average * (1 - variability), average * (1 + variability))`.

Funkce 1, triangularAV



Obr. 5, Grafické zobrazení funkce triangularAV

- `uniform(double min, double max)` – spojitá funkce, ohraničená z obou stran, vzorek leží v rozmezí [**min**, **max**]. Hustota pravděpodobnosti nezávisí na hodnotě x . Funkce se používá k výběru náhodné hodnoty, která může být v libovolně malém intervalu zvoleného rozmezí. Je důležité, že maximální hodnota má nulovou pravděpodobnost být vybranou.



$$f(x) = \frac{1}{\max - \min}$$

\min = minimum x
 \max = maximum x

Obr. 7, Popis funkce uniform

Obr. 6, Grafické zobrazení funkce uniform

3. Současný stav problematiky, modelování

Modelování v současnosti je řešením celé řady problémů souvisejících s projektováním nových a změnami v již funkčních procesech. Většinou problém není možné vyřešit pomocí reálných experimentů, protože jednak je to velice drahá varianta, která se může ve skutečnosti stát neziskovou anebo vůbec nefunkční, na druhou stranu některé změny jsou dokonce neuskutečnitelné. V takových případech máme k dispozici modelování, které umožňuje sestavit simulační model reálného systému s určitou úrovní abstraktnosti. Pod úrovní abstraktnosti se rozumí to, jak přesný bude model – jaké detaily budou považované za nepodstatné a nebudou popsány, a jaké části systému budou označeny jako důležité. Reálný systém je vždycky složitější než jeho model. Po zpracování simulačního modelu můžeme provést potřebné experimenty a testy, porovnat jejich výsledky pro získání optimálního řešení dosažení své cíle.

Existuje spousta nástrojů umožňujících modelovat různé systémy, například elektronické tabulky, ve kterých obvykle pracují s čísly, tím se modelují příjmy, náklady apod. Zároveň jsou i programy simulačního modelování, pomocí kterých můžeme vytvářet modely dynamických systémů jakékoliv složitosti. Může to být jak malá kancelář, tak i celé letiště anebo dokonce i vojenské akce atd.

Elektronické tabulky jako modelovací nástroj patří ke skupině analytického modelování. Analytické modelování je jedním ze způsobů sestavení matematického modelu. Takový princip popisuje proces pomocí některých funkcí (algebraických, diferenciálních, integrálních rovnic). Analytickou simulaci je možné použít pro popis dostatečně jednoduchých systémů, pokud je potřeba popsat komplikovanější proces, může nastat problém s matematickým popisem. Proto se obvykle snaží zjednodušit původní model, z čeho se vyskytují další problémy, nejdůležitějším z nich je to, že získané výsledky jsou jen přibližné. Analytický model ukazuje matematický vztah mezi vstupními a výstupními veličinami, ale neříká nic o vnitřní struktuře procesu. (2)

V některých případech nelze najít analytické řešení. Pro dynamické systémy není vhodné použít analytický model, protože existuje celá řada parametrů s nelineárním chováním a komplikovanými vzájemnými vztahy, které buď nelze popsat anebo popis je velice obtížný, a bylo by racionálně využít simulační modelování. Simulační model nám ukazuje, jak se mění stavy systému, podle jakých podmínek se to dělá, funkci systému jako celku a jeho dílčích částí. Podmínky přechodu mezi stavy můžou být popsány matematickými rovnicí, databází, diagramy, plány atd. Určitou výhodou simulačního modelu je názornost. Uživatelé jsou schopni vidět, jak funguje systém v čase, jaké procesy probíhají a jak dlouho to trvá. Existuje možnost přidat měřiče, aby bylo vidět, jak se mění parametry systému. Pokud nasimulovaný proces je homogenní, ve smyslu, že činí stejnou funkci, například obráběcí stroj určitého výkonu, je možné měnit parametry stroje a jeho vstupních veličin přímo v běžícím modelu. Takové možnosti práce se simulačním modelem dovolují využít velký rozsah příležitosti pro lepší prozkoumání systému.

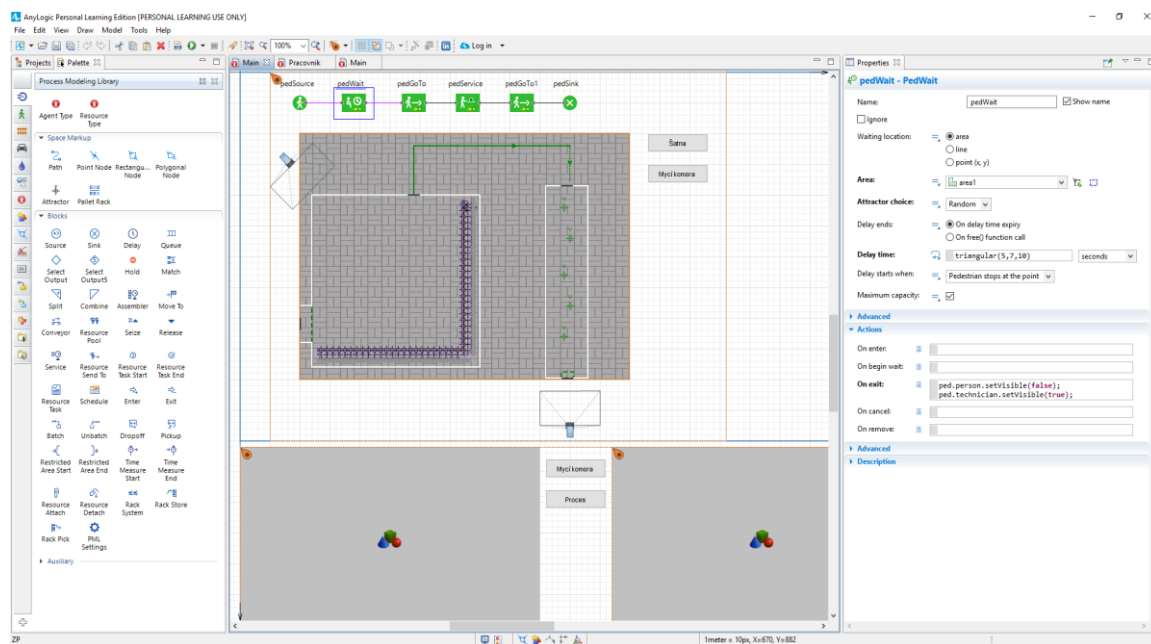
Simulační modelování má tři zásadní metody: (3)

- **Diskrétní simulace** – taková metoda představuje fungování systému jako posloupnost událostí. Každá událost vykonává v určitý čas a má vlastní popis funkci (matematický, programovací kód apod.), podle kterého se přepíná stav systému. Nejdůležitějšími částmi diskrétní simulace jsou: čas, událost či posloupnost událostí, doba čekání (provedení nějaké operaci), funkci pro generaci náhodných čísel pro zajištění chování modelu co nejbliž ke realitě, sběrnice statistických údajů, podmínky pro skončení simulaci. (4)
- **Systémová dynamika** – metoda zkoumání složitých systémů, která sleduje jejich chování v čase a v závislosti na vzájemném působení elementů systému (včetně kauzálních vztahů, zpožděných reakcí, vlivů prostředí, zpětných vazeb atd.). (5)
- **Multiagentní modelování** – takový typ sestavení modelu využívá agenty, které mohou představovat cokoliv a kohokoliv a mají své vlastní chování. Multiagentní modelování zkoumá, jak chování jednotlivých agentů (mikroelementů systému) ovlivňuje celý systém (makro úroveň). (6)

4. Návrh řešení – AnyLogic

Simulační model neboli „imitace“ vyžaduje speciální nástroje. Většinou se pro vytváření simulace komplikovaného procesu je potřeba mít program s pochopitelným rozhráním a jednoduchou strukturou, aby se uživatel mohl zamyslet jen nad tím, jak sestavit celý model, a ne nad tím, jak by on měl vymýšlet složité vztahy a kódy pro popis jednotlivých částí procesu. Jedním z nejlepších a nejrozšířenějších programů pro simulační modelování je AnyLogic. AnyLogic je výběrem velkých firem pro plánování rozpočtu, simulaci změn ve výrobních procesech a na trzích, demonstrace pro klienty apod.

AnyLogic – moderní simulační nástroj, který je schopen pracovat se všemi typy simulačního modelování. Program dovoluje pracovat s velkým instrumentářem. Pro sestavení modelu se využívají takzvané bloky – základní elementy programu, které mají široký rozsah funkcí pro vytváření simulačního modelu. AnyLogic využívá grafické uživatelské rozhraní (anglicky Graphical User Interface neboli GUI). Elementy takového rozhraní se představují uživateli na obrazovce jako obrázky. Výhodou takového rozhraní je názornost, jednoduchý přístup k formátování elementů, interface je „user friendly“ – lehce se naučí i začátečníci.



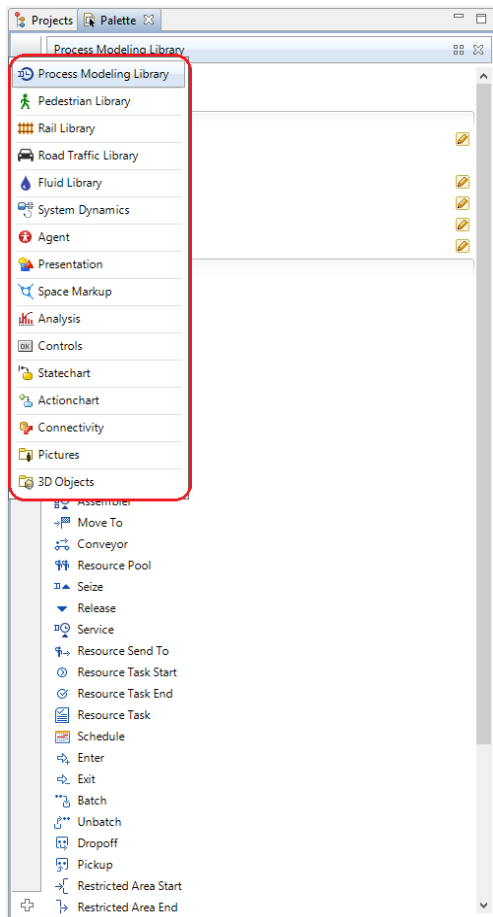
Obr. 8, Rozhraní AnyLogic

Uprostřed je pracovní oblast, kam se přemísťují bloky z palety, která je vlevo na obrázku. Vedle palety je menu „Projects“. Toto menu obsahuje nedávno otevřené modely ve formě rozbalovacího seznamu, který obsahuje veškeré bloky, objekty, agenty a jejich vazby, buď logické nebo fyzické, jako cesty určené pro pohyb agentů. Agenty různých modelů mohou mít stejný název. Dokonce existuje možnost překopírování objektů jednoho modelu na jiný, tím se zachovávají nastavené vlastnosti. V současném modelu lze používat agenty jiných otevřených modelů, tím je možné vyhnout zákazů použití více než 10 agentů v modelu, který platí ve verzi PLE, nicméně použití agentů jiného modelu je těžší ke zpracování, simulace může probíhat pomaleji, bude potřeba více RAM-paměti a možnost chyb se zvětšuje. Vpravo jsou vlastnosti elementu. Různé elementy mají odlišné vlastnosti

a parametry, které lze měnit. Základní vlastnosti objektu se zobrazují nahoře, k nim patří: jméno objektu, které je možné buď zobrazit nebo schovat, a nastavení zásadních parametrů, podle kterých blok funguje. Pro funkční bloky, které pracují s agenty, v seznamu „advanced“ se nacházejí parametry zpracování agentů, seznam „actions“ obsahuje podmínky, které říkají, kdy bude akce splněná, a řádek vedle, do kterého se píše akce v programovacím jazyce Java. Některé bloky obsahují několik seznamů „advanced“, obvykle, pokud je jich víc než jeden, seznam obsahuje nějaké specifické parametry, jako například kdy se přechází agent do nějaké zóny, v „advanced“ zadáváme tolerance vyjádřená v jednotkách délky, se kterou bude konečná zóna dosažená. Každý blok obsahuje seznam „description“, kde je možné popsat k čemu to slouží. Bloky určené pro prezentaci modelu obsahují parametry polohy a rozměru, poloha se zadává podle souřadnic a rozměr podle šířky, výšky a hloubky.

Při vytváření nového modelu se AnyLogic ptá, jaké jednotky modelového času chcete využít. Výběr jednotek se vyskytuje při zadání časových veličin některých bloků a dále při spouštění simulaci modelu. Simulace probíhá s rychlosti zvolené časové jednotky za sekundu. Volba jednotek času závisí na procesu, který je simulován. Například, je vhodné použít hodinu jako jednotku času při simulaci procesu výroby, pro dlouhodobé procesy, jako prodej zboží, rozšiřování epidemie nebo odhad finančního stavu v průběhu několika let, je vhodné použít dny, týdny, měsíce nebo roky, v některých případech je mnohem lepší využít sekundy, aby model probíhal co nejladněji a byla vidět hezká animace objektů.

AnyLogic obsahuje celou řadu nástrojů, proto je možné popsat stejný proces několika způsoby. Veškeré nástroje jsou umístěné podle účelu do skupin, které se nachází na takzvané paletě. V modelu je možné prohlédnout statistiku jednotlivých bloku, stisknutím levého tlačítka myši na bloku se zobrazí veškerá informace o jeho činnosti.



Obr. 9, Nástroje AnyLogic (1)

Process modeling library – elementy pro práce s procesy a agenty.

Pedestrian library – elementy pro práce s lidmi jako chodci. Agentem je člověk/skupina lidí.

Rail library – prvky pro železnice.

Road traffic library – prvky pro silniční dopravu.

Fluid library – prvky pro simulaci chování tekutin.

System dynamics – systémová dynamika.

Agent – elementy patřící k agentům.

Presentation – elementy pro prezentace a design modelu.

Space markup – vyznačení prostoru.

Analysis – elementy analýzy (data a jejich zpracování, tabulky, grafy apod.).

Controls – pomocné řídicí prvky.

Statechart – stavový diagram.

Actionchart – diagram akcí.

Connectivity – práce se soubory mimo AnyLogic.

Pictures – obrázky.

3D objects – 3D objekty.

Pro účely diskrétního modelování AnyLogic obsahuje veškeré potřebné k vytváření modelu elementy. Skupina Process Modeling Library je určena pro tento typ modelování. Model se vytváří jako grafický diagram procesu, elementy představují jednotlivé akce. Základní bloky používané k zpracování takového modelu jsou:



Obr. 10, Blok "source"

„Source“ je zdrojem agentů, obvykle je začátkem procesu. Základními parametry bloku jsou intenzita, podle které se generují agenti, maximální počet vygenerovaných agentů, a výběr agentu, v procesu používaných. Pokud je potřeba, aby agent měl zvláštní tvar nebo animace, je potřeba předem vytvořit agenta a pak ho použít v procesu.



Obr. 11, Blok "delay"

„Delay“ – blok zpožďuje agenty na zadanou dobu. Blok imituje provádění nějaké akce, procesu, obsluhy apod. Základním parametrem je doba zpoždění, která může být nastavená nejen číslem ale i funkcí, a kapacita agentů zpracovávaných najednou.



Obr. 12, Blok "queue"

„Queue“ – funkce bloku je taková: blok queue přijímá agenty z předchozího bloku a čeká, dokud ty nebudou přijaté blokem následujícím. Blok má nastavitelnou kapacitu, možnost zvolení jedné z variant čekací disciplíny (FIFO – anglicky *first in, first out*, což znamená první vstupuje, první vystupuje; LIFO – opačná varianta, anglicky *last in, first out*, což znamená, že vstupuje poslední a vystupuje první; podle priority, pokud je nastaveno příslušné rozlišení agentů; podle porovnání agentů). Při převýšení nastavené kapacity, nový vstupující do fronty agent může vytlačit poslední anebo nebude do fronty puštěn sám. Agenty, které se nedostali do fronty nebo byly vytlačené, mohou být zpracované za pomoci jiných bloku. Pokud je potřeba, je možné nastavit oblast, ve které agenty budou čekat.



Obr. 13, Blok "sink"

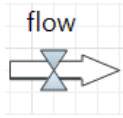
Element „sink“ vymazuje vstupující agenty. Obvykle je závěrečným bodem procesu. Pokud je potřeba zachovat agenty pro jiné operace, je potřeba využít blok „exit“ místo „sink“.

Základní bloky systémové dynamiky:



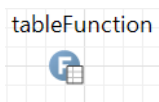
Obr. 14, Element "stock"

„Stock“ – jeden ze dvou fundamentálních elementů systémové dynamiky. „Stock“, neboli stav, reprezentuje procesy reálného světa, například sklad zboží, znalosti, lidé, peníze apod. Stav je statickou, akumulující částí systému, která se měří v čase a ukazuje množství existující v určité době.



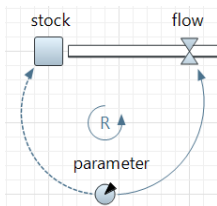
Obr. 15, Element "flow"

„Flow“, neboli tok, je druhým fundamentálním elementem systémové dynamiky. Tok na rozdíl od stavu není statickou částí procesu. Tok určuje tempo změny veličin souvisejících stavů – zadává dynamiku systému v čase.



Obr. 16, Element "table function"

„table function“ zadává funkci závislosti toků na stavů. Funkce může být jednoduchou rovnicí, která udává reakce toku, v závislosti na jednom nebo obojích připojených stavech.



Obr. 17, Ostatní elementy systémové dynamiky

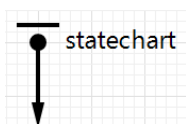
Parametry – jsou pomocné proměnné, obsahující informaci potřebné pro některou funkci.

Vazba mezi bloky vypadá jako šipka, bývá čárkovaná, v závislosti na připojených blocích.

Kruhový element uprostřed se jmenuje „loop“, a je jen grafickým objektem, který pomáhá pochopit směr vazeb systému.

Multiagentní modelování může využívat elementy určené především pro systémovou dynamiku nebo diskrétní simulace, ale jestli je potřeba, aby agent měl určitý stav, na kterém závisí jeho akce a odezvy, je možné využít elementy skupiny „statechart“, neboli stavový diagram.

Základními bloky stavového diagramu jsou:



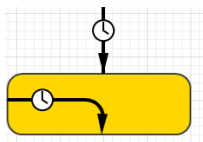
Obr. 18, Element
"statechart entry
point"

„Statechart entry point“ – element definuje počáteční stav diagramu. Pro každý stavový diagram musí být právě jeden „statechart entry point“. V parametrech lze nastavit akci, které se budou konat při přechodu kontroly nad agentem do počátečního stavu.



Obr. 19, Element
"state"

„State“ – stav. Stav reprezentuje řídicí bod, který obsahuje rádu odezev na splnění podmínek či akcí. Stav může být jak jednoduchý, tak i komplikovaný, kdy obsahuje více jednoduchých stavů. Kontrola je vždycky v jednom z jednoduchých stavů, ale současná odezva je sjednocením všech vnitřních jednoduchých a komplikovaných stavů. V parametrech je možné nastavit akci, které se budou konat při vstupu či výstupu ze stavu.

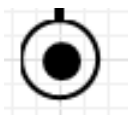


Obr. 20, Element
"transition"

„Transition“ – element ukazuje přechody mezi jednotlivými stavy anebo vnitřní přechody v jednom stavu. Přechod složí jako indikátor toho, že byly splněné určité podmínky, a stav diagramu se mění. Podle změny stavu se provedou nastavené akce. Při kreslení přechodu je potřeba správně spojit stavy mezi sebou – bod počátku a bod konce by měli ležet na elementu stav, když je to připojené správně, bod se vybarví zelenou barvou. Není chybou, kdy je čára nakreslena přes grafické zobrazení stavu, ale není s ním spojitá, důležitý je, aby program pochopil, jaké bloky jsou pomocí čáry spojené.

Možnost provedení přechodu může být vybraná v parametrech elementu. Možnosti jsou: zpoždění zadávané v jednotkách času; intenzita přechodu (množství za čas); splnění určité podmínky – booleovský výraz; získání specifikované zprávy – to může být text, číslo, booleovská hodnota, objekt, agent nebo výraz; příchod agentu.

Další nastavitelné možnosti jsou: akce prováděná při přechodu, bezpečnostní podmínka, která musí být splněná pro zajištění přechodu.

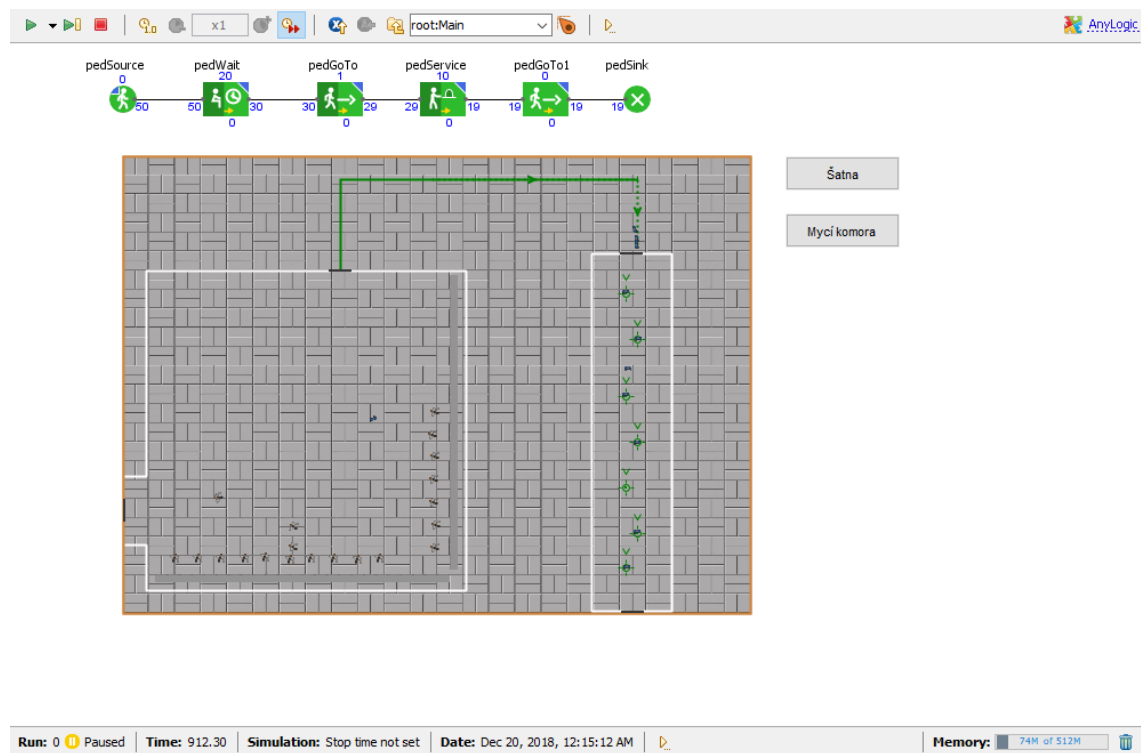


Obr. 21, Element
"final state"

„Final state“ – finální stav. Při přechodu do finálního stavu se provádí akce, v tomto elementu popsána, a stavový diagram se odstraňuje. Žádný přechod nemůže z finálního stavu vycházet.

Simulace se zapíná pomocí zeleného tlačítka na panelu vlevo nahoře nebo stisknutím tlačítka F5. V rozbalovacím seznamu tlačítka na panelu programu je možné zvolit simulaci nebo experiment ke spouštění.

Okno spuštěné simulace je na obrázku 22. Nahoře jsou hlavní ovládací tlačítka: spuštění s možností nastavení konečného času nebo nastavení času na který bude simulace spuštěná, krokovací tlačítko k přepínání modelovacího času po krocích, stop – zastavení simulaci a vynulování času; dále jsou tlačítka ovládání času: nastavení rychlosti na výchozí hodnotu, zmenšení rychlosti běhu simulace, zvětšení rychlosti běhu simulace, maximální rychlost běhu simulace – probíhá co nejrychleji do momentu zastavení; modré tlačítko s křížem uprostřed přemístuje pohled na plochu experimentu, červené tlačítko s písmenem „A“ přemístuje pohled na agent nejvyššího úrovně, v rozbalovacím seznamu jsou veškeré objekty modelu, výběrem elementu se zobrazí jeho vlastnosti, oranžová šipka s vlastním rozbalovacím seznamem obsahuje veškeré oblasti pohledu modelu.



Obr. 22, Okno simulace

5. Popis procesu

Podle Moorůva zákona (Gordon Earle Moore, spoluzakladatel firmy Intel) (7) „počet tranzistorů umístěných na integrovaném obvodu se zdvojnásobí každé dva roky“, který platí dodnes, je vidět, že tempo vývoje výroby polovodičů a integrovaných obvodu je velmi vysoké. V současné době existují hromadně vyráběné procesory podle technologií 7 nm, které jsou více výkonné a nízkenergetické, používají se v mobilních telefonech a tabletách. Z důvodu neustálého rozvoje, moderní továrny se snaží vylepšit své technologické procesy, všude, kde je možné, jsou používány roboty a rozmanité manipulátory pro zjednodušení lidské práci a získání co nejefektivnější výroby. V následujících bodech budou vysvětleny jednotlivé části celého procesu výroby polovodičů, popsány simulační modely každé z těchto částí a celkové výroby.

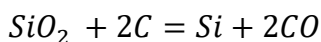
Celý proces výroby je rozdělen do několika částí: (8) (9)

5.1 „Clean-room“

Vstup do výrobních prostorů je oddělen. Před vstupem každý pracovník se má převleknout a projít komorou mytí vzduchem. Jsou požadovány speciální oděvy, boty, rukavice, obličejové masky. Personál prochází přes komoru s mytím vzduchem pro odstranění prachu a veškerých jiných částic. Celý výrobní prostor je takzvaný „Clean-room“ (čistý prostor), ve kterém se dodržují podmínky přísné čistoty, taková výroba vyžaduje natolik ideální čistotu, že uvnitř „Clean-room“ je čistší než v operační sále. V průběhu výroby jsou přesně sledované a regulované některé parametry, jako teplota prostoru, vlhkost, tlak, magnetické pole, elektrostatický výboj, vibrace a jiné. Zároveň se používá spousta vzduchových filtrů, potřebné pro výrobu čistá voda a jiné kapaliny jsou dodávány prostřednictvím trubic, umístěných přes celý prostor.

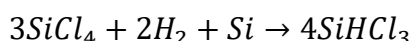
5.2 Výroba krystalického ingotu a křemíkových destiček

Základním materiálem pro výrobu polovodičů je křemíková destička. Křemík je druhým nejrozšířenějším prvkem v přírodě. Obsah křemíku v zemské kůře je 25-30%. Křemenný písek má velký podíl křemíku v podobě oxidu křemičitého. Pro výrobu je to základní surovina. Písek (SiO_2) je zkoksován v obloukové peci při teplotě $\sim 1800^\circ\text{C}$.



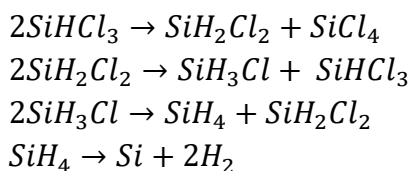
Vzorec 1

Vytvářený křemík je takzvaně „technický křemík“, který dosahuje 98-99.9% čistoty, ale pro výrobu polovodičů je potřeba použít „elektronický křemík“. V „elektronickém křemíku“ nesmí být víc než jeden cizí atom na miliard atomů křemíku. Za pomoci chlorací „technického křemíku“ se získává tetrachlorid křemíku (SiCl_4). V následujícím kroku tetrachlorid křemíku bude transformován do trichlorsilanu (SiHCl_3):



Vzorec 2

Následující chemické reakce jsou využívány k dalšímu čištění křemíku. Za pomoci recyklace sekundárních obsahujících křemík prvků reakcí se sníží cena výroby a vyřeší některé ekologické problémy.



Vzorce 3, 4, 5, 6

Tak se získávají „elektronický křemík“ s čistotou skoro 100% (99.99999%). V nádobu s roztaveným křemíkem, která se otáčí, se spouští tyč rozměrem jako tužka otáčecí na opačnou stranu. Pomocí regulace rychlosti otáčení a teploty, z nádoby se pomalu vytahuje válcový krystalický ingot. Rychlost vytažení je zhruba 1.5 mm/min. Rozměry ingotu dosahují výšky člověka a váhy v 100 kilogramů. Vytažený křemíkový ingot musí být upraven na jednotný průměr. Po kontrole upraveného ingotu, ten je připraven k rozřezání do tenkých destiček. Přes dopování destiček jinými elementy jako As+, P+, B+ se změnil vodivost a charakteristiky křemíku. Povrch a okraje destiček jsou pak vyleštěny a vybroušeny, po tomto kroku už jsou připravené k následující výrobě integrovaných obvodů.

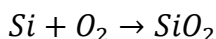
5.3 Design

Prvním krokem výroby integrovaných obvodů je vytvoření designu pomocí speciálního softwaru počítače. Design slouží k tomu, aby vzory každého integrovaného obvodu byly připravené. Tyto vzory jsou pak nanášené na foto masku pomocí elektronových paprsků či laseru. Počet masek potřebných pro výrobek s integrovaným obvodem obvykle závisí na složitosti designu a výrobní technologii. Většinou je potřeba nejméně 20 až 30 vrstev foto masek, které musí být velmi přesně mezi sebou zarovnané.

5.4 Diffusion area (oblast difúze)

V této oblasti křemíkové destičky jsou přemísťované do peci ve formě trubice pro narůstání tenké vrstvy (~1 μm) za zvýšených teplot (315°C–320°C). Tenká vrstva je vrstvou oxidu křemičitého (SiO₂), který slouží jako ochrana od dopadu jiných prvků. Možnost vytváření oxidové vrstvy dělá křemík jedním z nejvíc používaných prvků pro polovodičové substráty.

Na začátku destičky procházejí speciálním čištěním. Křemíková destička je po čištění sušena, a na určitou dobu je rozmístěna v ohříváné křemenné trubici, kde teplota a průtok plynu jsou přesně kontrolovány. Do trubici se dodává kyslík vysoké čistoty anebo směs kyslíku a dusíku. Chemická reakce pro vytváření dioxidové vrstvy je následující:



Vzorec 7

Je to takzvaná „suchá“ oxidace.

5.5 Photolithography area (oblast fotolitografie)

V této oblasti je zajištěn přenos schémat obvodu. Tenká vrstva fotorezistu (fotorezistivní kapalina) se jednotně nanáší na povrch destičky, tím je rozmístěna foto maska. Světlo dopadající na destičku přes foto masku tvoří obraz exponovaných a

neexponovaných oblastí v závislosti na foto masce. Neexponované oblasti se pokrývají fotorezistivní vrstvou. Celý proces vypadá takhle:

- **Čištění a příprava povrchu**

Křemíková desky se čistí v ultrasonické komoře nebo koupeli s různými organickými rozpouštědly – aceton, metanol a oplachování v isopropanolu. V případě značných znečištění je potřeba použít směs kyseliny sírové a peroxidu vodíku ($H_2SO_4 + H_2O_2$) s následujícím použitím procesu RCA čištění. RCA čištění je standardní proces pro křemíkové desky potřebný k provedení před operací vyžadující vysokou teplotu. Proces se skládá z: odstranění organických látek, odstranění oxidové vrstvy (volitelný krok, závisí na technologii), odstranění iontového znečištění.

- **Nanesení fotorezistu**

Nejčastěji používaná metoda je odstředění. Taková metoda dovoluje vytvářet homogenní film fotorezistu při přesné kontrole rychlosti otáčení (obvykle několik tisíc otáček za minutu).

- **Předběžné sušení**

Po nanášení fotorezistu je potřeba provést předběžné sušení. Proto se deska přesouvá do peci na několik minut při teplotě 100-120°C. Tento krok se využívá k odpařování rozpouštědla v obsahu fotorezistu, což je vhodné ke zlepšení adheze, vyloučení lpění k foto masce, možnosti nanášení další vrstvy fotorezistu.

- **Expozice**

Proces expozice spočívá v zasvěcení fotorezistu přes foto masku s potřebnou šablonou pomocí viditelného světla nebo ultrafialového záření. Zároveň v tomto spočívá rozdíl fotolitografie a jiných typů litografie – při fotolitografii se využívá vlastnosti světla, při jiných typech litografie se využívá rentgenové záření, ionty či elektrony.

Hlavními parametry expozice jsou: vlnová délka, doba expozice a výkon zdroje záření. Každý fotorezist má jinou veličinu dávky (mJ/cm^2) potřebné k expozici, proto je důležité správně navrhnout zdroj. Pokud bude dávka nepostačující, může nastat problém projevu fotorezistu, kdyby naopak dávka byla nadměrná fotorezist může být poškozen. V závislosti na množství vyráběných desek za hodinu (anglicky wph – wafer per hour) se rozdělují fotolitografické stroje.

- **Projev**

V průběhu projevu se část fotorezistu odstraňuje pomocí speciální kapaliny – vývojky (například tetramethylamoniumhydroxidu), tím se vytvářejí „okna“ ve fotorezistivní vrstvě. V případě použití pozitivního fotorezistu se odstraňuje exponovaná oblast, nebo naopak při negativním fotorezistu. Určitě vývojka se vybírá podle typu fotorezistu.

- **Obrábění povrchu**

5.6 Ion implantation area (oblast implantace iontů)

Proces implantace iontů se používá pro změnu vodivosti materiálu. Nabité částice se implantují do konkrétní oblasti křemíkové destičky. Změna vodivosti se provádí kontrolou koncentrace iontů, hloubky dopadu – energie, teplotou – čím vyšší je, tím rychleji probíhá redistribuce nosičů, a dobou konání operace. Proces iontové implantace probíhá ve vakuu, a částice jsou urychlované elektrickým polem několika tisíc voltů. Částice potřebné hmoty se vybírají pomocí magnetického pole.

Cíl takové operace – změnit typ vodivosti křemíku a koncentrace nosičů v objemu polovodiče pro získání potřebných parametrů p-n přechodu. Nejrozšířenějšími datovacími prvky pro křemík jsou fosfor a arsen – získání vodivosti n-typu, a bor pro vodivost p-typu.

Po příčině, že taková operace by měla být prováděna ve vakuu, implantace iontů může trvat 2 až 3 hodiny, protože před tím než začít, je potřeba dosáhnout hlubokého vakuu v komoře, kde se ta operace provádí.

5.7 Chemical vapor deposition area (oblast chemické depozice z plynné fáze)

Chemické reakce nastávají v reakční komoře. Chemical vapor deposition neboli CVD – proces při kterém na povrchu křemíku narůstá vrstva dodatečného materiálu. Takový typ depozice se využívá k vytváření tenké vrstvy pomocí srážení molekul za vysokých teplot (900–1300°C). Plyn obsahující křemík a přísady „p-“ a „n-“ typu (obvykle vodík) se propouští nad ohřátými deskou, takhle narůstá vrstva křemíku s potřebnými přísady. Proces obvykle probíhá při atmosférickém tlaku.

Základní typy plynné epitaxe:

| | |
|--|---------------------------------------|
| Interakce vodíku s tetrachloridem křemičitým (1150-1300°C) | $SiCl_4 + 2H_2 \rightarrow Si + 4HCl$ |
| Pyrolytické srážení silanu (1000-1100°C) | $SiH_4 \rightarrow Si + 2H_2$ |
| Interakce vodíku s trichlorsilanem | $SiHCl_3 + H_2 \rightarrow Si + 3HCl$ |
| Rozklad trichlorsilanu | $SiH_2Cl_2 \rightarrow Si + 2HCl$ |

Tab. 1

5.8 Etching area (oblast leptání)

Zde se provádí leptání exponovaných oblastí (nepokrytých fotorezistivní vrstvou). Zbývající obraz – je prostor potřebný pro integrovaný obvod. Ve výrobě se využívá leptání pro výrobu speciálních mikročipů: proto aby neprovádět proces fotolitografie několikrát v každém cyklu, vrstvy potřebné pro fotolitografie se nanáší najednou a pak se postupně provádí proces projevu. Tím se zmenší počet chemických obrábění, čas potřebný k nanášení vrstev (vyčerpání komory trvá ~2-2.5 hodiny) a také cenu výrobku. Po leptání zbývá oblast potřebná k vytváření obvodů.

5.9 Physical vapor deposition area (oblast fyzikální depozice z plynné fáze)

Z předchozích oblastí destička je obsažená několika tisíci až několika miliony elektronických prvků. Tyto prvky musí být propojené vodivými metalickými drátky, aby mohli plnit svoji funkci. Zde povrchově odleptána destička se pokrývá metalickou vrstvou pomocí rozprašování. Proces nanášení tenké vrstvy je zajištěn pomocí fotolitografie a odleptání pro odstranění nepotřebných částí. Tento proces nechává metalické dráty, které propojují každý elektronický prvek. Existují tři základní metody depozice:

- **E-paprsková metoda:** zaměřený elektronový paprsek se používá k ohřívání metalického materiálu. Proud elektronů je urychlován v elektrickém poli v rozmezí 5-10 kV. Zaměřený paprsek taví materiál nacházející v komoře s vodním chlazením. Roztavený metal se odpařuje a kondenzuje na povrchu chladné desky a celém prostoru komory. Dále následují procesy fotolitografie a leptání, po kterých se získává komplikované elektrické schéma.
- **Pulsní odpařování:** tato metoda se provádí, když je potřeba, aby slitina dvou materiálů se odpařila najednou. Příklady takových slitin – chrom + nikl (chromnikl, v závislosti na typu slitiny obsahuje 55-78% niklu a 15-23% chromu), chrom + oxid křemičitý (SiO) a silumin (slitina hliníku a křemíku). Při pulsním odpařování se provádí pomocí tepelného odporu a nepřetržitě napájené cívky. Roztavené metaly se pak přenáší do povrchu desky.
- **Rezistivní neboli katodové odpařování:** nejjednodušší a nejlevnější z procesů. Odpařování je zajištěno pomocí postupného zvětšení elektrického proudu procházejícího katodou, tím se materiál roztaví a značně pokrývá katodu. Proud katodou se zvětšuje, pokud se z ní neodpaří metal.

5.10 Chemical mechanical polishing area (oblast chemicko-mechanického leštění)

V této oblasti se využívá principu mechanického obrábění a chemických reakcí k efektivnímu odstranění materiálů pro získání ploché destičky připravené k následujícímu naprašování tenkých vrstev.

Veškeré složité a velmi přesně prováděné procesy jsou opakovaně prováděny ke zpracování integrovaných obvodů. Každá destička se skládá ze stovek až tisíc čipů.

Po ukončení všech procesů mikročipy procházejí kontrolami a testy, až tyto prohlídky jsou provedené, destičky budou rozřezané na jednotlivé čipy, které se připraví k dalším testům podle účelu konkrétního integrovaného obvodu.

6. Simulační model továrny

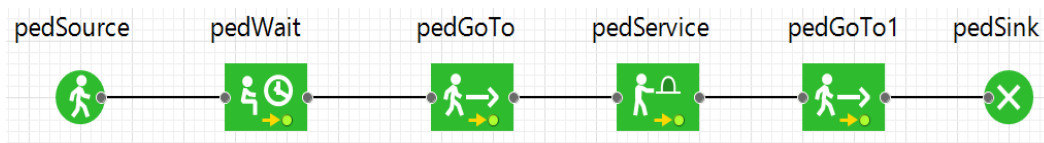
Pro lepší pochopení model celé továrny jsem rozdělil do několika částí podle jednotlivých bodů vysvětlených v předchozí kapitole. Simulační model představuje sebou zjednodušený přehled funkcí a procesů v čase. Pro tuto model jsem zvolil sekundy jako jednotky času pro simulaci, přesto že spousta výrobních procesu probíhá v rozmezí od několika minut do několika hodin, sekundy jsou zvolené pro správné a plynulé zobrazení chování pracovníků.

Model továrny bude provedená pomocí všech třech typů simulačního modelování, v závislosti na procesu budou použité různé agenty. Agent může být čímkoliv a kýmkoliv, jak člověkem, tak i, například, objednávkou anebo výrobkem.

6.1. Model vstupu pracovníků do čistého výrobního prostoru

V této části je využít typ diskrétní simulace, jako agenty vystupují pracovníci přicházející do práce. Takový typ byl zvolen kvůli elementům vhodným pro popis následujícího modelu. Důležitějším elementem je blok wait/delay, který imituje vykonání nějakých akcí s agenty, v tomto případě – převlekání pracovníků a průchod komorou mytí vzduchem. Proces je popsán s využitím bloků skupiny pedestrian library.

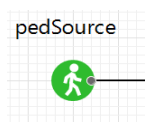
Pro simulace příchodu pracovníků byla zvolená následující varianta:



Obr. 23, Model vstupu do továrny

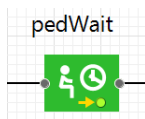
Tato varianta byla vyřizena pomocí elementů pedestrian library pro jednodušší představu. Elementy pedestrian library mají zelenou barvu a ve většině případů se shodují s elementy process modeling library, ale, jak je to vidět mají prefix „ped“. AnyLogic PLE umožňuje simulace 1 hodiny při práci s bloky pedestrian library, což v našem případě stačí.

Jako agent tohoto procesu byl vytvořen nový agent „Pracovník“ a vybrané vhodné 3D objekty pro animaci. Při vytváření agentů se AnyLogic ptá, jestli to bude jeden agent (dodavatel, budova, sklad, továrna, nemocnice apod.), nebo celá populace agentů, pracujících ve stejném prostředí (lidé, spotřebitele, auta, zboží apod.). V našem případě byla zvolena populace agentu, které představují pracovníky připravující se k nástupu do výroby.



Obr. 24, Element „pedSource“

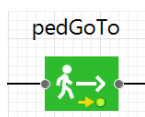
„Source“ je základním elementem pro diskrétní simulace. Tento blok generuje agenty, které budou vykonávat různé činnosti v modelu. Pro ukázkou byla zvolená intenzita generace agentů – příchodu pracovníků 200 za hodinu a maximální počet agentů – 100. Agenty jsou generované v oblasti „area“.



Obr. 25, Element "pedWait"

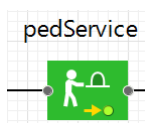
Blok „wait“ nebo „delay“ simuluje vykonání nějaké činnosti, v našem případě blok „pedWait“ představuje čas, kdy se pracovníci převlékají. Doba čekání je zadána pomocí funkce `triangular(4, 5, 6)` [minuty]. V parametrech bloku byla nastavena akce při výstupu agentů - `ped.person.setVisible(false); ped.technician.setVisible(true);`. V tomto kódu `ped` představuje agenta, který je v procesu použit, `person` a `technician` – 3D objekty tvaru agentu, `setVisible` – nastavit viditelnost. To znamená, že pracovníci se převlékají do speciálního oděvu, a počáteční model pracovníku se mění na jinou.

Pro takový blok je možné zvolit místo čekání/činnosti. Proto byla vytvořena oblast „area1“ s vygenerovanými elementy typu „attractor“ pro určení správného místa a směru pohledu agentů. Agenty vybírají místo náhodně podle volného attractoru. Attractor představuje sebou určitý bod v nějaké oblasti.



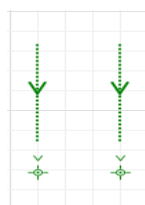
Obr. 26, Element "pedGoTo"

Blok „GoTo“ zajišťuje přechod agentů do zvolené oblasti, bodu podle jeho souřadnic anebo přechod podél čáry. Při přechodu podél čáry je možné zvolit přesnost podle zadávání maximální vzdálenosti od ní při průchodu. Pro tuto model byl vybrán přechod podél čáry „targetLine1“ z šatny do mycí komory. Přesnost dosažení čáry je nastavena na 1 metr.



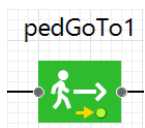
Obr. 27, Element "pedService"

Element „service“ může být použit místo „wait“ a „delay“, protože má shodnou funkci, ale někdy je vhodné použít blok „service“ místo jiných. „pedService“ pracuje s elementy „service with area“ nebo „service with lines“, které dovolují nastavit kolik bude míst obsluhy a front před nimi, fronty může nebýt, v takovém případě se využívá zónu pro čekání. Zároveň je možné vybrat chování fronty, například jako příští pro obsluhu může být zvolen agent, který je nejbližší ve frontě nebo agent v nejkratší frontě, anebo více prioritní agent, pokud je nastaveno takové rozlišení.



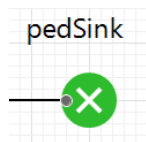
Obr. 28, Fronty

Pro mycí komoru byl zvolen případ s jednou frontou a sedmi místy pro obsluhu – vlastně mytí vzduchem. Následující agent vstupuje do komory jen když je volné místo pro vykonání potřebných činnosti, pokud vše místa jsou obsazené, agent čeká před vchodem. Doba obsluhy je nastavená na 2 minuty.



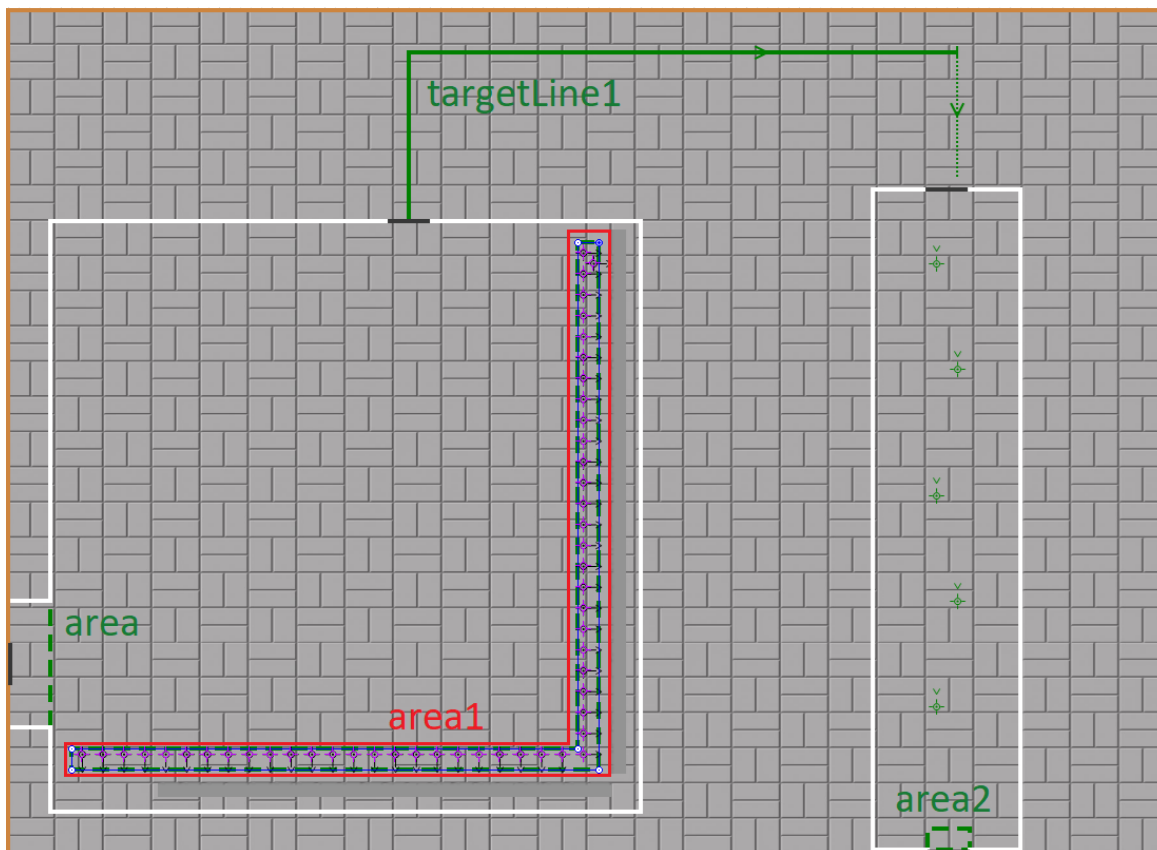
Obr. 29, Element "pedGoTo1"

Element „pedGoTo1“ zajišťuje agentům průchod komorou a vstup do výrobního prostoru. Jako bod přechodu byla zvolená oblast „area2“.



Element „Sink“ vymazává vstupující agenty, obvykle je finálním blokem procesu. V našem případě simuluje vstup pracovníku do výrobního prostoru a vystup z oblasti této části modelu.

Obr. 30, Element
"pedSink"



Obr. 31, Oblast modelu vstupu do továrny

K zobrazení 3D simulaci byly použité elementy skupiny „Presentation“ – „3D window“, který představuje oblast se zobrazením pohledu z kamery, a „camera“, která se podle souřadnic nastavuje na určité místo v modelu a předává obraz na „3D window“.

Pro přepínání místa pohledu na model, byly použité 3 elementy „View Area“, které umožňují změnit pohled na jeden z předem zvolených. Aby zajistit jednoduchý přechod mezi oblastmi prohlížení, funkcí přechodů byly zapsané do akcí tlačítek. Kód těchto akcí je následující: `viewArea.navigateTo()`, kde `viewArea` – odpovídá názvu zvolené oblasti prohlížení a `navigateTo()` – je akce, která říká „přemístit se ke zvolené oblasti“.

Jen pro ukázkou byly použité elementy zobrazující stěny a podlahu. Element „wall“ (česky stěna) se nachází ve knihovně pro popis chodců. Stěna se používá nejen pro dekorační účely, stěna omezuje prostor pro pohyb chodců. Tak v modelu vstupu do továrny, stěny omezují místnosti a žádný pracovník nemůže jít nikam jinam než do oblasti vynechané pro dveře. Podlaha byla udělaná pomocí elementu „“ knihovny prezentace. Pro takové elementy, které není potřeba měnit v průběhu modelování, je možné ve vlastnostech nastavit blokování. Zablokované elementy nelze vybrat stisknutím tlačítkem myši. Kdyby

bylo potřeba jeden z nich vymazat či upravit, je potřeba to najít v příslušných elementech běžného projektu.

6.2. Model procesu výroby polovodičů pro mikroprocesory

Pro tuto část simulačního modelu za jednotku modelovacího času jsou zvolené hodiny. Nicméně, rychlost pohybujících se objektů je popsána v metrech za sekundu, stejně parametry naložení a vykládky předmětu do a z regálů jsou zadané v sekundách. Proto k prohlednutí možností 3D animace modelu je možné změnit časové jednotky v parametrech modelu. Vlastnosti modelu se otevírají výběrem potřebného modelu v menu projekty vlevo od palety.

Celý proces výroby polovodičových systémů pro mikroprocesory je popsán v bodě 4. V moderních procesorech se opakují kroky 5, 6, 7, 8 a 9: fotolitografie, implantace iontů, chemická depozice z plynné fáze, leptání a fyzikální depozice z plynné fáze. Tyto kroky zajišťují zpracování jedné masky. Podle složitosti procesoru a zvoleného technického provedení se liší počet opakování. Jako výrobní měřítko se často používá „dny pro zpracování jedné vrstvy“. Obvykle nanesení jedné vrstvy se trvá od 1 do 1,5 dnů, nejlepší továrny – 0,8 dnů. Procesor vyrobený podle technologií 28nm má 40 až 50 masek, 14nm/10nm – 60 vrstev, 7nm – 80 až 85 vrstev. S technologií 5 nm se očekává 100 masek. (10)

Ze známého času pro zpracování jedné vrstvy a počtů vrstev je možné spočítat, jak dlouho koná výroba jednoho procesoru od začátku do konce. Po vybudování nové továrny ona má 4 roky k tomu, aby zkompenzovat vložené peníze (více než 5 miliardu dolaru) a začít produkovat zisk. Bylo spočítáno, že pro splnění požadavků je potřeba vyrábět aspoň 100 správně pracujících desek za hodinu. Jako vyráběný procesor byl zvolen moderní Apple A12 Bionic, který je používán v iPhone XR a XS. Procesor je vyráběn pomocí technologií 7 nm, má architekturu ARMv8.3 a obsahuje 6,9mld tranzistorů s logikou CMOS. Rozměr jednotlivých čipů na křemíkové desce je: 9,89 mm X 8,42 mm, což odpovídá 83,27 mm². Ve výrobě se používají desky o průměru 12", z čeho je lehce spočítáno, že na jedné desce se nachází víc, než 800 čipů. (11)

$$12" = 30.48 \text{ cm}$$

$$S_{deska} = \frac{\pi * \text{průměr}^2}{4} = \frac{\pi * 30.48^2}{4} = \frac{\pi * 929.0304}{4} = 729.66 \text{ cm}^2$$

$$\text{počet čipů} = \frac{S_{deska}}{S_{čipu}} = \frac{729.66 * 10^2 \text{ mm}^2}{83.27 \text{ mm}^2} = 876,26$$

Vzorce 8, 9, 10

Z důvodu, že čipy jsou obdélníkové a deska je kulatá, nemůže být celá plocha desky obsazena mikročipy.

Pro simulace továrny byl zpracován model, obsahující veškeré potřebné k výrobě kroky a opakováním procesu nanesení integrálních obvodu 85krát. Hlavními agenty modelu jsou desky, pro které byl vytvořen zvláštní agent s 3D grafikou. Každý agent reprezentuje jednu desku. Dalšími agenty modelu jsou jednotlivé kroky procesu výroby: Diffusion, Photolithography, IonImplantation, CVD (chemical vapor deposition), Etching,

PhysicalVaporDeposition, ChemicalMechanicalPolishing a agent pro automatické manipulátory továrny – AutoManipulator. Každý z nich simuluje provádění své činnosti.

Auto-manipulátory přemísťují desky do strojů a ze strojů do skladovacích regál. Regály jsou použité jako prostor pro čekání následující operaci. Různé stroje zachycují desky na různou dobu, proto je potřeba mít místo skladování desek čekajících na obsluhu. V simulačním modelu byla zvolena varianta využít regály patřící jednotlivým procesům. Auto-manipulátory mají tři parkoviště mezi kterými oni jsou rozdělené rovným dílem. Tři parkovacích oblasti jsou vytvořené v různých místech k tomu, aby zkrátit dobu, kterou stráví manipulátory při pohybu. Vždy se vybírá nejbližší k cílovému místu manipulátor. Počet potřebných manipulátorů je závislý na počtu pracujících strojů, počet manipulátorů je zadán jako počet strojů krát 10, kde 10 představuje počet korků v jednom cyklu vynásobený dvěma – jeden pro dodání a jeden pro výběr desek ze stroje.

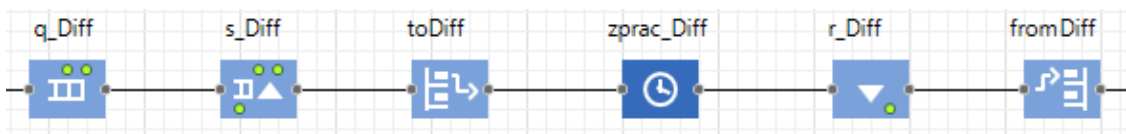
Proces výroby se začíná blokem source, který najednou generuje zvolený počet desek ke zpracování v určené k tomu oblasti modelu. Vygenerování desek lze nastavit na potřebnou rychlost ve vlastnostech objektu source. V modelu se desky generují najednou nastavením rychlosti rovné počtu desek v milisekundu. Při nastavení vzniku desek na pomalejší hodnotu, budou nově vygenerované desky přepravené pomocí volných auto-manipulátorů. Maximální možný počet desek pro model je 100. K tomu, aby provést simulaci ve větším rozsahu, je potřeba změnit počet buněk nebo úrovní regálů. Většinou AnyLogic dovoluje zapisovat číselné hodnoty pomocí parametrů nebo proměnných, ale pro objekt reprezentující regály je zápis ve formě odkazů zakázán. Pokud vzhled modelu není důležitý nebo úroveň abstraktnosti je vysoká, je možné předem nastavit velký počet buněk regálů. Nicméně, ve svém modelu jsem chtěl nejen ukázat velkou řadu funkcí simulačního systému. Cílem bylo ukázat, že některé procesy je možné sestavit ve 3D grafice, dá se takovým způsobem jednoduše vysvětlovat technologické procesy, které vyskytují v běžném studiu. Pro změnu počtu desek slouží parametr „pocet_desek“, který je možné upravit před spuštěním simulaci.

Veškeré agenty kromě agentu „desky“ jsou prostředky vykonávající různé činnosti. Proto je potřeba použít element „ResourcePool“, který zadává vlastnosti agentů, k nim patří: počet agentů; jejich místo; definice, jestli je agent pohyblivý nebo statický; rychlost pro pohyblivé agenty; konec směny podle rozvrhu; přestávky, havárie, obsluhu podle rozvrhu nebo distribuční funkci; speciální úkoly. Každý prostředek má 5 bodů, ve kterých se nacházejí stroje. Důležitý je to, že pro správnou činnost je potřeba zachytit statické prostředky, které pracují s hlavním agentem. K tomu slouží element seize – blok zachycuje prostředky ke zpracování vstupujících agentů. Ve vlastnostech je potřeba zvolit prostředek, který bude zachycen. Je také možné nastavit frontu a její kapacitu. V modelu jsou kapacity všech objektů typu seize nastavené pomocí parametru „pocet_stroju“. Fronty a určité kapacity jsou nastavené z důvodu, že model bude obsahovat konečný počet strojů, fronta v daném případě imituje čekání vzorku na uvolnění místa pro další krok zpracování. Po každém zachycení prostředku a následujícím provedení zvolené činnosti je potřeba jeho uvolnit, jinak nastane chyba. K uvolnění slouží element release.

Pro každý stroj byl zvolen 3D objekt z knihovny AnyLogic pro zobrazení v modelu. Pro některé objekty jako jsou stroje, existuje dvě varianty zobrazení – otevřený (volný) nebo uzavřený (pracující). Imitace práce stroje zajištěná pomocí kódu: isIdle() nebo

isBusy(). Agent má 2 objekty k zobrazení, ve vlastnostech objektu je možné nastavit jejich viditelnost, tak u agentu reprezentujících stroje viditelnost objektů je nastavená následujícím způsobem: isIdle() pro otevřený stroj, funkce isIdle() nabývá hodnoty true, když stroj je volný, a isBusy() pro zavřený stroj, funkce nabývá hodnoty true, když je zachycen agent reprezentující stroj.

Každý proces má své blokové schéma. Blokové schéma obsahuje: pomocný blok queue, který obsahuje agenty, dokud neuvolní místo ve frontě elementu seize (s_*název procesu*); seize zachycuje zvolený prostředek pro následující krok procesu; pomocí elementu rackPick jsou desky přemísťované ze skladovacího regálu do stroje, ve vlastnostech je nastavena cíl, kam se přestěhuje agent – v našem případě je to zachycený prostředek, a přesun pomocí auto-manipulátoru; element delay imituje obrábění desky; release uvolňuje zachycený agent; rackStore zajišťuje přesun desek do dalšího skladovacího regálu.



Obr. 32, Blokové schéma jednoho z výrobních procesů

Doba obsahu pro elementy typu „delay“ imitujících provedení procesu je zadána pomocí funkce triangularAV s odchylkou 1%. Jednotlivé doby byly nastaveny jen přibližně tak, aby každý z opakovaných cyklu trval 24 hodiny. Procesy implantace iontů a leptání trvají 4 hodiny, fotolitografie – 2 hodiny, procesy vyžadující vysoké vakuum (CVD, PVD) trvají delší – 7 hodin. Procesy difuze a chemicko-mechanického leštění jsou nastavené na 3 a 2 hodiny.

Výrobní technologie vyžaduje nanesení určitého počtu vrstev, proto aby v modelu byly opakované jen vybrané kroky je potřeba zajistit správnou zpětnou vazbu. K oddělení opakujících se procesu byl vybrán element „select output“. Takový blok má jeden vstup a dva výstupy: výstup True a výstup False. Element vybírá výstup podle zvolené pravděpodobnosti 0 až 1 nebo podle podmínky. Když je podmínka splněná agent prochází výstupem True, když podmínka není splněná, výstup True je zablokovan a výstup False odblokovan. V modelu jsou vytvořené pomocné parametry „pom_prom_cyklus“ pro načítání agentů procházejících blok rackStore procesu PVD, který je posledním krokem opakujícího se cyklu, a „pom_prom_cyklus1“ pro načítání čísla cyklu, je zadán následujícím kódem:

```
if ((pom_prom_cyklus)%(pocet_desek) == 0){
pom_prom_cyklus1 + +; }
```

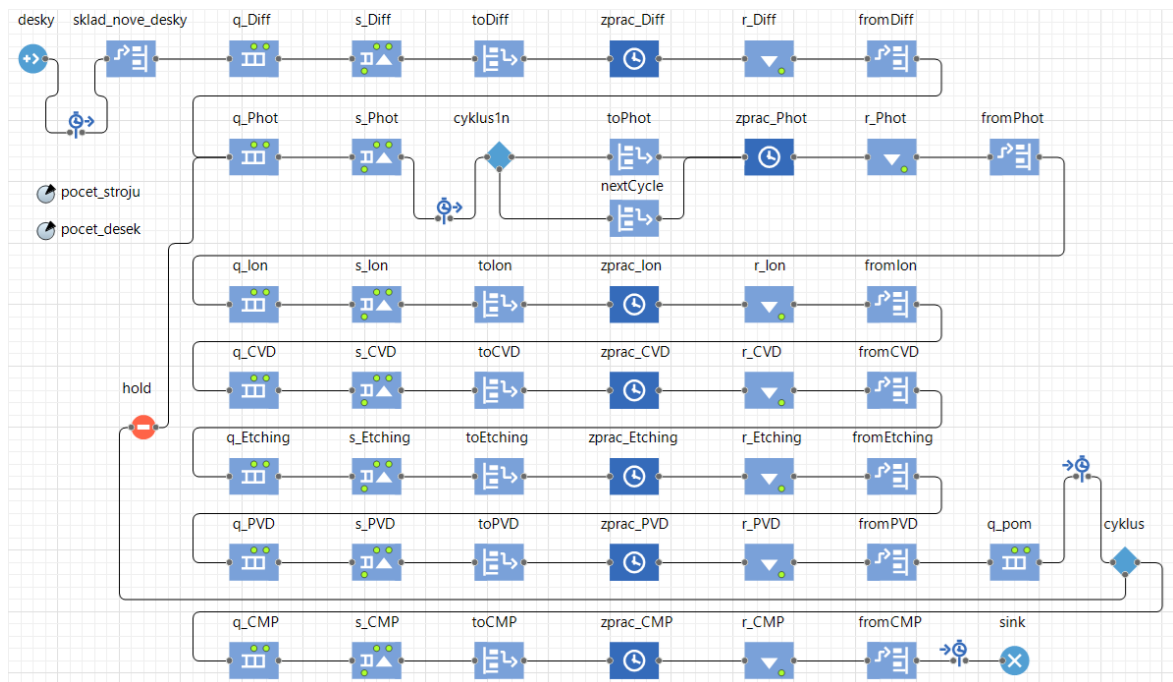
Kód 1

To znamená že, když je zbytek divize nulový, pomocný parametr se načítá. Pomocí těchto parametrů je vidět, kolik agentů prošlo krokem PVD a jaký cyklus se provádí současně.

V modelu jsou použité 2 elementy typu „select output“: „cyklus“ a „cyklus1n“. Blok „cyklus“ má podmínku: (pom_prom_cyklus >= (pocet_desek * 84 + 1)), tím je

zajištěno, že cyklus bude opakován 85krát před otevřením výstupu True. Blok „cyklus1n“ má podmínku: $(pom_prom_cyklus1 == 0)$, která nabývá hodnoty True jen v prvním cyklu.

Blokové schéma celého výrobního procesu je na následujícím obrázku:



Obr. 33, Blokové schéma celého výrobního procesu

Jak je vidět ze schématu a popisu bloků „cyklus“ a „cyklus1n“, desky generované v elementu source prochází procesem difuze a přechází do procesu fotolitografie, kde blok „cyklus1n“ použít agenty do výstupu True. Agenty jsou pak vyzvednuté z regálu „storage_Photo“, který je určen jako místo pro skladování desek po procesu difuze. Po průchodu cestou dalších zpracování, agent vstupuje do bloku „cyklus“, který ověřuje kolik agentů prošlo předchozím krokem a otevírá výstup False dokud nebudou agenty procházet posledním výrobním cyklem. Zpětná vazba pro další cyklus následuje červený blok „hold“. Tento element blokuje průchod pro vstupující agenty. Výstup může být odblokován několika způsoby podle nastavení: pomocí funkcí block() nebo unblock(); automatické blokování po vstupu n agentů, pro odblokování je potřeba využít funkci unblock(); podle podmínky, která může být určena pro každý agent. Přesto že element „select output“ může blokovat agenty sám, bylo potřeba využít blok „hold“. Problém je v tom, že první agenty budou procházet posledním krokem opakovaného cyklu rychleji než poslední z generovaných. Tím bude způsobena chyba, která spočívá v tom, že agenty přicházející na další výrobní cyklus vstupují do elementu, který je hlídá v regálu, určeném pro desky procházející prvním cyklem, když agent nebude nalezen v regálu, nastane chyba a simulace se ukončí. Proto slouží element „hold“ který zadržuje agenty, dokud neprojdou první etapou veškeré vygenerované desky. Element „hold“ je blokován na začátku simulaci. Pro blokování a odblokování jsou vytvořené pomocné parametry „pom_prom_block“ a „pom_prom_unblock“. Parametr pro odblokování se načítá v elementu „q_pom“, který slouží jako pomocný objekt, obsahující agenty vstupující do bloku „cyklus“, zabrání chybám nedostatku kapacity pro agenty, protože „select output“ nemůže v sobě obsahovat

agenty. Když je načítaný parametr se rovná počtu generovaných agentů, element „hold“ se odblokuje. Blokovací parametr se načítá v elementu „q_Phot“. Kdy se parametr rovná počtu desek vynásobeného počtem cyklu, které je potřeba provést, blok „hold“ se blokuje znovu.

Důležitými parametry takové výroby jsou střední čas zpracování jedné vrstvy a střední doba celé výroby jedné desky s mikročipy. Pro zjištění časových parametrů jsou bloky měření času:



„Time measure start“ – blok začíná měřit čas, kdy přes ni prochází agent.

Obr. 34, Time measure start



„Time measure end“ – blok zapisuje a zpracovává data získané průchodem agentů. Databáze bloku obsahuje informace o průchodu všech agentů, spočítává střední čas průchodu od začátku měření času, střední kvadratickou odchylku a jiné veličiny. Detaily jsou v simulace modelu po stisknutí pravého tlačítka na objektu skončení měření času nebo jednotlivé parametry můžou být zapsané do proměnné či textového bloku pomocí funkcí.

Obr. 35, Time measure end

Pro zápis těchto parametrů bylo vytvořené proměnné: „doba1_cyklu“ a „doba_zpracovani“. Model obsahuje dva páry bloků měření času pro zjištění potřebných parametrů. Doba jednoho cyklu se zapisuje po průchodu agentů výstupem True bloku cyklus. K zajištění správného zápisu střední hodnoty je použit pomocný parametr a zápis se provádí, když objektem prošli veškeré vygenerované agenty. Správná funkce je zajištěná pomocí kódu:

```
pom_prom_doba_cyklu ++;
```

```
if (pom_prom_doba_cyklu == pocet_desek){
    doba_1cyklu = timeMeasureEnd.distribution.mean();}
```

Kód 2

V kódu je odkaz na konkrétní parametr potřebného bloku měření času, v daném případě – timeMeasureEnd.distribution.mean() vrací střední hodnotu ze všech změřených.

Střední doba celé výroby jedné desky s mikročipy je zapsána do proměnné „doba_zpracovani“ pomocí parametru „pom_prom_doba_zpracovani a “ stejného kódu, který se vykonává po průchodu všech vygenerovaných agentů do úplně posledního bloku modelu – „sink“.

Simulace nemá nastavený určitý čas skončení, protože při změně parametru se mění doba výroby ve velkém rozsahu, který většinou nelze vypočítat. Z toho důvodu je zvolená varianta pro ukončení simulace pomocí programových kódů. Pro tuto funkci byla vytvořené pomocná proměnná „pomstop“, která se zvětšuje při výstupu agentů z bloku

fromCMP, který je posledním blokem, který vykonává činnost s agenty. Po průchodu všech vygenerovaných agentů se simulace ukončí.

```
pomstop ++;
```

```
if (pomstop == pocet_desek){  
    finishSimulation();}
```

Kód 3

Pro účely průzkumu byla nastavená obsluha pro pracující stroje. Obsluha se nastavuje následujícím způsobem: program se ptá, za jak dlouho přijde první žádost o obsluhu, čas do další obsluhy, úroveň priority (určen pro případ s několika různými akcí, rozlišuje pořadí jejich vyplnění), povolení „vyhánět“ pro daný typ obsluhy (akce s vyšší úrovní priority může vyhánět jiné akce z fronty), existuje možnost výběru typu obsluhy – zpožděním nebo podle diagramu, a nastavení, jak se zapisuje do statistiky čas ztracený na obsluhu stroje.

Obrázek 36 ukazuje nastavení obsluhy pro obráběcí stroje. Zapínání obsluhy se provádí pomocí parametru „obsluha“ typu boolean, který je možné nastavit před provedením simulaci. Čas do první žádosti o obsluhu zadán pomocí funkci uniform s hodnotami od středního času provedení jednoho procesu daného stroje do dvojnásobku této hodnoty. Čas do další obsluhy je zadán pomocí funkci triangularAV se střední hodnotou rovnající dvojnásobku času provedení jednoho procesu a procentuální odchylkou 5%. Priorita v daném případě není potřebná, protože žádné jiné akce zastavující stroje nejsou. Čas potřebný k obsluze je stejný pro vše stroje a je zadán pomocí funkci triangularAV se střední dobou obsluhy – 30 minut, a odchylkou 30%. Statistika se nesbírá v průběhu odstávky a obsluhy stroje.

| | | |
|------------------------------|--|---------|
| Maintenance: | <input type="checkbox"/> | obsluha |
| Initial time to maintenance: | <input type="text" value="uniform(10000, 21600)"/> | seconds |
| Time to next maintenance: | <input type="text" value="triangularAV(21600, 0.05)"/> | seconds |
| 'Maintenance' priority: | <input type="text" value="100"/> | |
| 'Maintenance' may preempt: | <input type="checkbox"/> | |
| Maintenance type: | <input type="text" value="Delay"/> | |
| Maintenance time: | <input type="text" value="triangularAV(1800, 0.3)"/> | seconds |
| Usage statistics are: | <input type="text" value="not collected"/> | |

Obr. 36, Obsluha agentu

Model obsahuje 9 kamer ukazujících důležitější místa: oblast prvního parkoviště manipulátoru, dodávky surovin a cesty do regálu pro jejich skladování, oblasti obráběcích strojů, regál pro hotové desky a oblast expedice. Pomocné tlačítka zajišťují jednoduché přepínání mezi zvolenými oblastmi pohledu, k nim patří: pohled na model shora, pohled každé z devíti kamer a samotný proces s příslušnými bloky. Diagram kromě popisujících proces bloků obsahuje využití pomocné proměnné, proměnné ukazující střední dobu trvání jednoho cyklu a celou dobu zpracování zvoleného počtu desek, histogram ukazující procentní rozdělení desek podle doby trvání jednoho cyklu.

7. Experiment

Experimenty AnyLogic slouží k tomu, aby získat přesnější data z několika ukončených simulací vytvořeného modelu. Parametry se mění automaticky ve zvoleném rozsahu nebo náhodně. Obvykle v úkolech návrhu simulačního modelu je potřeba docílit určité přesnosti, proto je potřeba provést simulaci více krát. V úlohách se nastavují doby zpracování nebo obsluhy pomocí distribučních funkcí s větší odchylkou. Proto při provedení experimentu se získávají odlišné výsledné hodnoty, ze kterých se pak spočítává střední hodnota, která by měla odpovídat stanovenému zadání. Pro některá zařízení je potřeba nastavovat určitý nebo náhodný čas pro obsluhu, odstranění havárií nebo nechávat čas po každém provedeném procesu k zotavení stroje. Takovým způsobem spolu s nastavením doby zpracování pomocí funkcí se docílí dostatečně přesných výsledků. K výpočtu potřebného čísla experimentů se využívá metody teorii pravděpodobnosti, například teorém Moivre-Laplace (12). Provedením obsluhy či jiné potřebné akcí se omezuje počet současně pracujících strojů, tím vzniká fronta a zvětšuje se střední doba zpracování.

Pro model výroby polovodičů pro mikroprocesory byl zvolen experiment se změnou počtů pracujících strojů (viz Obr. 36). Cílem je zjistit, jak počet strojů ovlivňuje střední dobu jednoho cyklu a střední dobu zpracování jedné desky. Počet strojů se mění automaticky v rozsahu od 1 stroje každého typu do 40 strojů. Rozsah se nastavuje ve vlastnostech vytvořeného experimentu. Pro zobrazení získaných hodnot byly vytvořené proměnné odpovídající časovým veličinám.

Zápis se provádí pomocí funkcí:

```
if (root.pocet_stroju == N){  
    stredni_doba_cykluN = root.doba_1cyklu;}  

```

Kód 4

```
if (root.pocet_stroju == N){  
    stredni_doba_zpracN = root.doba_zpracovani;}  

```

Kód 5

Kde N odpovídá počtu strojů. Pokud v kódu je odkaz na objekt nacházející na jiném agentu, je potřeba v kódu ukázat ke kterému agentu patří, například: zde jsou používány proměnné agentu „Main“, který je hlavním agentem modelu, proto je před odkazem na proměnnou ukázán agent, ke kterému to patří, zde root odpovídá agentu nejvyššího úrovně – „Main“.

Před spouštěním experimentu je možné změnit počet desek k vygenerování, veškeré ostatní veličiny se změny podle nastaveného počtu agentů. Jediným problémem je omezení maximálního počtu desek. Maximálně je možné nastavit 100 desek, pro nastavení většího počtu je potřeba změnit počet buněk nebo úrovní regálů. K nastavení počtu desek slouží proměnná „nastaveni_poctu_desek“, která ve vlastnostech experimentu odkazuje na parametr „počet_desek“.

K zapínání obsluhy slouží vytvořená proměnná „nastaveni_obsluha“, kterou je možné změnit před spouštěním experimentu pomocí elementu „checkbox“.

Maximální hodnota použité RAM-paměti je cca 1300 Mb, k tomu, aby experiment měl rezervu, je ve vlastnostech experimentu maximální dostupná paměť nastavena na 4096 Mb.

Čas skončení experimentu je nastaven v samotném modelu. Experiment provádí veškeré akce vytvořené v modelu při každé změně parametrů a spouštění dalšího pokusu.

Střední doba jednoho cyklu je nejdůležitějším parametrem ke zjištění. Čas zpracování pro každý proces je nastaven tak, aby v součtu jeden cyklus trval průměrně 24 hodiny. Výsledky experimentů se mění v závislosti na nastavených hodnotách. Střední hodnoty získané v experimentu se mohou měnit při nastavené větší odchylce v distribuční funkci zadávací čas potřebný k provedení jednotlivých procesů. Určitě se mění i čas při zapnutí obsluhy strojů, tím by se měli měnit veškeré časové veličiny.

Ze získaných hodnot je vidět, že střední doba trvání jednoho cyklu s přesností +/- 1 hodina se docílí 24 hodin při použití aspoň 29 strojů pro každý proces. Podstatné zmenšení pokračuje do pokusu s 34 pracujícími stroje, následující změny počtu strojů skoro nemění střední dobu provedení jednoho cyklu. Získané hodnoty určitě platí jen pro zvolený počet desek. V tomto případě bylo využito 100 desek. Při zmenšení počtu desek se střední doba cyklu docílí 24 hodin při menším počtu pracujících strojů. V případě se zapnutou obsluhou strojů může nastat situace, kdy je střední doba jednoho cyklu delší přesto, že počet pracujících strojů je větší, to je způsobeno tím, že podle distribučních funkcí nastavených v parametrech obsluhy se náhodně mění časové hodnoty v dostatečně velkém rozsahu.

Vytvořený experiment je vhodné použít pro jakýkoliv rozsah surovin a strojů. Ke zvětšení počtu desek je potřeba provést změnu kapacity regálů a zvolit větší interval ve vlastnostech experimentu. K získání potřebných hodnot pro zvětšený rozsah strojů je potřeba vydělit vše proměnné na ploše experimentu najednou a několikrát okopírovat, čísla jednotlivých proměnných se změní automaticky. Pro správný zápis hodnot do nových proměnných je nutné upravit kód, který se vyplňuje po provedení experimentu, doplněním řádku s potřebnými čísly.

8. Závěr

Cílem závěrečné práce bylo vysvětlit principy modelování a popsat simulační prostředí AnyLogic. Základní myšlenkou bylo vysvětlit princip podle kterého se uživatel ovládá programem a vytváří vlastní model. K tomu bylo potřeba přidat popis typu modelování, které jsou současně používány v běžném životě. Zaprvé bylo popsáno rozhraní simulačního prostředí. Podle typu modelování byly rozepsané jednotlivé skupiny elementů a jejich pracovní bloky. Pro praktickou ukázkou byl zvolen proces výroby polovodičů pro mikroprocesory.

V průběhu přípravy závěrečné práce byl vytvořen simulační model výroby v prostředí AnyLogic PLE. Celý proces byl rozdělen do dvou modelů, kvůli omezení verze PLE na vytváření více než 10 agentů. Model vstupu pracovníků do výrobního prostoru byl vyplněn pomocí knihovny modelování chodců, protože činnými agenty modelu jsou lidé, a cílem bylo ukázat nejen samotný proces, ale i animace pracovníků. Model továrny byl vyplněn většinou pomocí knihovny modelování procesu.

Správnost simulace modelů byla ověřena v několika testech se změnou počátečních parametrů. Modely fungují správně pro zadané rozsahy hlavních veličin. Vytvořený experiment ukazuje jednu z nejdůležitějších funkcí simulačního modelování v prostředí AnyLogic. Experiment s automatickou změnou parametrů umožňuje rychle prozkoušet vliv změn na výsledné veličiny. V případě takového modelu lze z výsledných hodnot posoudit potřebný počet strojů k co nejrychlejšímu obrábění zvoleného počtu křemíkových desek a získat důležitou hodnotu trvání jednoho procesu nanášení vrstvy polovodičů.

Výhodou AnyLogic je lehce pochopitelné rozhraní a objektově orientovaný princip modelování. Užitečnou funkcí je možnost popisu objektů pomocí programátorského jazyku Java, který dodává celou řadu dalších možností. Experimenty AnyLogic dovolují provést velký počet simulací ve zkrácený čas a získat odlišující se výsledky ze kterých lze vybrat nejlepší variantu pro daný model. AnyLogic obsahuje velký instrumentář elementů k provedení jakéhokoliv z typů moderního modelování, kromě toho existuje možnost kombinovat různé přístupy k vytvoření modelu pro nejpřesnější popis zvoleného procesu. Pro některé procesy je potřeba vytvořit animaci pohybu objektu, pro takový případ AnyLogic má spoustu 3D objektů pro skoro jakýkoliv proces. Pomocí elementu kamera lze předem zvolit místo, které bude při simulaci prohlíženo. Správnou volbou modelovacího času a vlastností bloku je možné vytvořit úplně animovaný proces, který umožňuje nejen získat číselné údaje, ale i vizuálně ukázat celý proces. Podstatnou výhodou pro studenty je verze PLE, která je úplně zadarmo ke stažení.

Nevýhody AnyLogic ale také jsou. Velkou nevýhodou je omezení verze PLE, kvůli všem omezením někdy je potřeba vymýšlet úplně jiný postup ke zpracování nebo rozdělovat model na menší části, který budou pak vytvořené v několika modelech. Některé funkce jsou zablokované, proto uživatel musí využít svůj tvůrčí přístup k tomu, aby vyhnout zablokovaným funkcím a udělat potřebnou část pomocí nástrojů, které jsou k dispozici. Některé hodnoty nelze měnit pomocí parametrů, což je problémem pro rychlou změnu potřebné veličiny. Vždy je potřeba přesně spojovat elementy, protože kdyby agent neměl určité cesty, bude simulace ukončená kvůli chybě. Stejně se uživatel musí dávat pozor na grafické elementy, jako jsou cesty pro pohyb agentů. Když dvě cesty se nachází

ve stejném místě, může pohybující se agent ztrácet více času kvůli průchodu jednou z nich několikrát. V některých případech je obtížné vybrat několik elementů stejného typu pro změnu nějakých vlastností, program dovoluje jen ruční výběr každého elementu, nebo výběr všech objektu v zachycené oblasti, problém nastává, když je v oblasti hodně elementů různých typů.

Modelování se současně používá ve mnoha různých oblastech jak výroby, tak i plánování či průzkumu jiných typů prací a procesů. V moderních procesech využití modelování je nezbytné. Pomocí simulačních programů lze provádět testy a experimenty, které by byly velmi obtížné k probraní ve skutečnosti. AnyLogic za použití velkého rozsahu pracovních elementů dovoluje popsat libovolný proces. Tento program je výběrem velkých firem v celém světě. Svou bakalářskou práci jsem chtěl ukázat příležitostí, které AnyLogic má, a názorně vysvětlit, jak lze vytvářet modely různých procesů. Simulační modelování by mohlo být dobrým doplňkem v některých předmětech bakalářského či magisterského studia. Modelování je vhodné využít nejen k plánování změn procesů, ale i k vysvětlení samotného procesu buď velké výroby jako celku anebo rozděleného a rozepsaného po částech.

Seznam použitých informačních zdrojů

1. AnyLogic, Manuály.

2. Grigoryev, Ilya. *AnyLogic in Three Days: Modeling and Simulation Textbook*. 2014.

3. **Imitační modelování.**

https://ru.wikipedia.org/wiki/%D0%98%D0%BC%D0%B8%D1%82%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5. [Online]

4. **Dickrétní simulace.**

https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D1%81%D0%BA%D1%80%D0%B5%D1%82%D0%BD%D0%BE-%D1%81%D0%BE%D0%B1%D1%8B%D1%82%D0%B8%D0%B9%D0%BD%D0%BE%D0%B5_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5. [Online]

5. **Systémová dynamika.**

https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%BD%D0%B0%D1%8F_%D0%B4%D0%B8%D0%BD%D0%B0%D0%BC%D0%B8%D0%BA%D0%B0. [Online]

6. **Multiagentní modelování.**

https://ru.wikipedia.org/wiki/%D0%90%D0%B3%D0%B5%D0%BD%D1%82%D0%BD%D0%BE%D0%B5_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5. [Online]

7. **Moorův zákon.** https://cs.wikipedia.org/wiki/Moor%C5%AFv_%D0%B4%D0%B8%D0%BD%D0%B0%D0%BC%D0%B8%D0%BA%D0%B0. [Online]

8. David G. Baldwin, James R. Rubin, Afsaneh Gerami. *Производство кремниевых полупроводниковых приборов*. str. 246.

9. **Semiconductor Technology at TSMC, 2011.**
https://www.youtube.com/watch?v=4Q_n4vdyZzc. [Online] 2011.

10. **Corporation, © Intel. От песка до процессора.**
<https://habr.com/company/intel/blog/110234/>. [Online]

11. **A12 Bionic - Apple.** <https://en.wikichip.org/wiki/apple/ax/a12>. [Online]

12. **Локальная теорема Муавра — Лапласа.**
https://ru.wikipedia.org/wiki/%D0%9B%D0%BE%D0%BA%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%82%D0%B5%D0%BE%D1%80%D0%B5%D0%BC%D0%B0_%D0%9C%D1%83%D0%B0%D0%B2%D1%80%D0%B0_%E2%80%94%D0%9B%D0%B0%D0%BF%D0%BB%D0%B0%D1%81%D0%B0. [Online]

13. **Chip Manufacturing Process - Philips Factory.**
<https://www.youtube.com/watch?v=gBAKXvsaEiw>. [Online]

14. **GLOBALFOUNDRIES Sand to Silicon.**
<https://www.youtube.com/watch?v=UvluuAIiA50>. [Online]

15. **How a CPU is made.** <https://www.youtube.com/watch?v=qm67wbB5GmI>. [Online]
16. **Wafer fab.** <https://www.youtube.com/watch?v=dPh-mQ6irls>. [Online]
17. **Silicon Wafer Production.** <https://youtu.be/AMgQ1-HdElM>. [Online]
18. **Fabrication and Morphology of Polysilicon Layers.** <http://www.iue.tuwien.ac.at/phd/puchner/node32.html#SECTION00921000000000000000>. [Online]
19. **How they make silicon wafers and computer chips.** <https://www.youtube.com/watch?v=6rfUoMA-dco>. [Online]
20. **Corporation, © Intel. Где производят процессоры Intel.** <https://habr.com/company/intel/blog/109339/>. [Online]
21. **Трудности производства процессоров.** <https://habr.com/company/intel/blog/108615/>. [Online]
22. **LAPEDUS, MARK. Battling Fab Cycle Times.** <https://semiengineering.com/battling-fab-cycle-times/>. [Online]
23. **Semiconductor Manufacturing Techniques.** <https://www.allaboutcircuits.com/textbook/semiconductors/chpt-2/semiconductor-manufacturing-techniques/>. [Online]
24. **Пятковский, Юрий. Производство микрочипов: как это работает.** <https://itc.ua/articles/proizvodstvo-mikrochipov-kak-eto-rabotaet/>. [Online]
25. **Коленченко, Олег. Технологии полупроводников.** <https://www.ferra.ru/review/techlife/semiconductor-technology-part-1.htm>. [Online]
26. **Shilov, Anton. TSMC: First 7nm EUV Chips Taped Out, 5nm Risk Production in Q2 2019.** <https://www.anandtech.com/show/13445/tsmc-first-7nm-euv-chips-taped-out-5nm-risk-in-q2>. [Online]
27. **7 nanometer.** https://en.wikipedia.org/wiki/7_nanometer. [Online]
28. **Doping (semiconductor).** [https://en.wikipedia.org/wiki/Doping_\(semiconductor\)](https://en.wikipedia.org/wiki/Doping_(semiconductor)). [Online]
29. **Photolithography.** <https://en.wikipedia.org/wiki/Photolithography>. [Online]
30. **Chemical-mechanical polishing.** https://en.wikipedia.org/wiki/Chemical-mechanical_polishing. [Online]
31. **Plasma etching.** [https://en.wikipedia.org/wiki/Etching_\(microfabrication\)#Plasma_etching](https://en.wikipedia.org/wiki/Etching_(microfabrication)#Plasma_etching). [Online]
32. **Physical vapor deposition.** https://en.wikipedia.org/wiki/Physical_vapor_deposition. [Online]
33. **Chemical vapor deposition.** https://en.wikipedia.org/wiki/Chemical_vapor_deposition. [Online]
34. **Ion implantation.** https://en.wikipedia.org/wiki/Ion_implantation. [Online]

35. RCA clean. https://en.wikipedia.org/wiki/RCA_clean. [Online]

36. Боев, В. Д. **МОДЕЛИРОВАНИЕ в AnyLogic, Пособие для практических занятий.** Санкт-Петербург : Моделирование в AnyLogic. Пособие для практических занятий. — СПб.: ВАС, 2016. – 412 с., 2016. **ВОЕННАЯ АКАДЕМИЯ СВЯЗИ.**

ZavernaPrace2 : ParametersVariation

Spustit

Iterace: 40

Parametry

Počet strojů 40

Počet desek 100

Počet manipulátorů 400

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|--|--|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|--|--|---|--|--|--|--|--|
| <input checked="" type="checkbox"/> V stredni_doba_cyklus1 713.686 hodin | <input checked="" type="checkbox"/> V stredni_doba_cyklus6 119.447 hodin | <input checked="" type="checkbox"/> V stredni_doba_cyklus7 102.437 hodin | <input checked="" type="checkbox"/> V stredni_doba_cyklus8 89.711 hodin | <input checked="" type="checkbox"/> V stredni_doba_cyklus9 79.861 hodin | <input checked="" type="checkbox"/> V stredni_doba_cyklus10 71.507 hodin | <input checked="" type="checkbox"/> V stredni_doba_cyklus11 65.412 hodin | <input checked="" type="checkbox"/> V stredni_doba_cyklus12 60.016 hodin | <input checked="" type="checkbox"/> V stredni_doba_cyklus13 55.439 hodin | <input checked="" type="checkbox"/> V stredni_doba_cyklus14 51.544 hodin | <input checked="" type="checkbox"/> V stredni_doba_cyklus15 48.055 hodin | <input checked="" type="checkbox"/> V stredni_doba_cyklus16 45.164 hodin | <input checked="" type="checkbox"/> V stredni_doba_cyklus17 42.468 hodin | <input checked="" type="checkbox"/> V stredni_doba_cyklus18 40.117 hodin | <input checked="" type="checkbox"/> V stredni_doba_cyklus19 38.092 hodin | <input checked="" type="checkbox"/> V stredni_doba_cyklus20 36.138 hodin | <input checked="" type="checkbox"/> V stredni_doba_cyklus21 34.506 hodin | <input checked="" type="checkbox"/> V stredni_doba_cyklus22 32.864 hodin | <input checked="" type="checkbox"/> V stredni_doba_cyklus23 31.567 hodin | <input checked="" type="checkbox"/> V stredni_doba_cyklus24 30.372 hodin | <input checked="" type="checkbox"/> V stredni_doba_cyklus25 29.094 hodin | <input checked="" type="checkbox"/> V stredni_doba_cyklus26 27.954 hodin | <input checked="" type="checkbox"/> V stredni_doba_cyklus27 26.946 hodin | <input checked="" type="checkbox"/> V stredni_doba_cyklus28 26.058 hodin | <input checked="" type="checkbox"/> V stredni_doba_cyklus29 25.231 hodin | <input checked="" type="checkbox"/> V stredni_doba_cyklus30 24.682 hodin | <input checked="" type="checkbox"/> V stredni_doba_cyklus31 24.449 hodin | <input checked="" type="checkbox"/> V stredni_doba_cyklus32 24.344 hodin | <input checked="" type="checkbox"/> V stredni_doba_cyklus33 24.32 hodin | <input checked="" type="checkbox"/> V stredni_doba_cyklus34 24.23 hodin | <input checked="" type="checkbox"/> V stredni_doba_cyklus35 24.228 hodin | <input checked="" type="checkbox"/> V stredni_doba_cyklus36 2.068.051 hodin | <input checked="" type="checkbox"/> V stredni_doba_cyklus37 2.068.126 hodin | <input checked="" type="checkbox"/> V stredni_doba_cyklus38 2.069.048 hodin | <input checked="" type="checkbox"/> V stredni_doba_cyklus39 2.067.746 hodin | <input checked="" type="checkbox"/> V stredni_doba_cyklus40 2.066.976 hodin |
|---|---|---|--|--|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|--|--|---|--|--|--|--|--|

V nastaveni_poctu_desek
100

V nastaveni_obsluha
true

Obsluha

Run: 80 Finished Experiment: 100% Simulations: Stop time not set

Memory: 71M of 4,096M

112.6 sec

Obr. 37, Experiment ParametersVariation se zapnutou obsluhou

Tovarna : ParametersVariation

Spustit

Iterace: 40

Parametry

Počet strojů 40

Počet desek 100

Počet manipulátoru 400

| | | | | | | | |
|--|--|--|--|--|--|--|--|
| <input checked="" type="checkbox"/> stredni_doba_cyklus1 693.702 hodín | <input checked="" type="checkbox"/> stredni_doba_cyklus6 116.137 hodín | <input checked="" type="checkbox"/> stredni_doba_cyklus11 63.602 hodín | <input checked="" type="checkbox"/> stredni_doba_cyklus16 43.883 hodín | <input checked="" type="checkbox"/> stredni_doba_cyklus21 33.554 hodín | <input checked="" type="checkbox"/> stredni_doba_cyklus26 27.178 hodín | <input checked="" type="checkbox"/> stredni_doba_cyklus31 24.348 hodín | <input checked="" type="checkbox"/> stredni_doba_cyklus36 24.217 hodín |
| <input checked="" type="checkbox"/> stredni_doba_zprac1 59.249.112 hodín | <input checked="" type="checkbox"/> stredni_doba_zprac6 9.916.895 hodín | <input checked="" type="checkbox"/> stredni_doba_zprac11 5.431.512 hodín | <input checked="" type="checkbox"/> stredni_doba_zprac16 3.740.344 hodín | <input checked="" type="checkbox"/> stredni_doba_zprac21 2.886.511 hodín | <input checked="" type="checkbox"/> stredni_doba_zprac26 2.332.349 hodín | <input checked="" type="checkbox"/> stredni_doba_zprac31 2.080.287 hodín | <input checked="" type="checkbox"/> stredni_doba_zprac36 2.068.093 hodín |
| <input checked="" type="checkbox"/> stredni_doba_cyklus2 347.158 hodín | <input checked="" type="checkbox"/> stredni_doba_cyklus7 99.593 hodín | <input checked="" type="checkbox"/> stredni_doba_cyklus12 58.339 hodín | <input checked="" type="checkbox"/> stredni_doba_cyklus17 41.306 hodín | <input checked="" type="checkbox"/> stredni_doba_cyklus22 32.06 hodín | <input checked="" type="checkbox"/> stredni_doba_cyklus27 26.217 hodín | <input checked="" type="checkbox"/> stredni_doba_cyklus32 24.305 hodín | <input checked="" type="checkbox"/> stredni_doba_cyklus37 24.208 hodín |
| <input checked="" type="checkbox"/> stredni_doba_zprac2 29.648.344 hodín | <input checked="" type="checkbox"/> stredni_doba_zprac7 8.504.132 hodín | <input checked="" type="checkbox"/> stredni_doba_zprac12 4.982.15 hodín | <input checked="" type="checkbox"/> stredni_doba_zprac17 3.628.324 hodín | <input checked="" type="checkbox"/> stredni_doba_zprac22 2.739.096 hodín | <input checked="" type="checkbox"/> stredni_doba_zprac27 2.240.342 hodín | <input checked="" type="checkbox"/> stredni_doba_zprac32 2.076.311 hodín | <input checked="" type="checkbox"/> stredni_doba_zprac37 2.067.48 hodín |
| <input checked="" type="checkbox"/> stredni_doba_cyklus3 231.754 hodín | <input checked="" type="checkbox"/> stredni_doba_cyklus8 87.253 hodín | <input checked="" type="checkbox"/> stredni_doba_cyklus13 53.892 hodín | <input checked="" type="checkbox"/> stredni_doba_cyklus18 39.05 hodín | <input checked="" type="checkbox"/> stredni_doba_cyklus23 30.707 hodín | <input checked="" type="checkbox"/> stredni_doba_cyklus28 25.359 hodín | <input checked="" type="checkbox"/> stredni_doba_cyklus33 24.289 hodín | <input checked="" type="checkbox"/> stredni_doba_cyklus38 24.206 hodín |
| <input checked="" type="checkbox"/> stredni_doba_zprac3 19.971.223 hodín | <input checked="" type="checkbox"/> stredni_doba_zprac8 7.460.449 hodín | <input checked="" type="checkbox"/> stredni_doba_zprac13 4.602.698 hodín | <input checked="" type="checkbox"/> stredni_doba_zprac18 3.395.881 hodín | <input checked="" type="checkbox"/> stredni_doba_zprac23 2.643.609 hodín | <input checked="" type="checkbox"/> stredni_doba_zprac28 2.187.105 hodín | <input checked="" type="checkbox"/> stredni_doba_zprac33 2.074.722 hodín | <input checked="" type="checkbox"/> stredni_doba_zprac38 2.066.796 hodín |
| <input checked="" type="checkbox"/> stredni_doba_cyklus4 173.904 hodín | <input checked="" type="checkbox"/> stredni_doba_cyklus9 77.578 hodín | <input checked="" type="checkbox"/> stredni_doba_cyklus14 50.109 hodín | <input checked="" type="checkbox"/> stredni_doba_cyklus19 37.06 hodín | <input checked="" type="checkbox"/> stredni_doba_cyklus24 29.446 hodín | <input checked="" type="checkbox"/> stredni_doba_cyklus29 24.688 hodín | <input checked="" type="checkbox"/> stredni_doba_cyklus34 24.221 hodín | <input checked="" type="checkbox"/> stredni_doba_cyklus39 24.204 hodín |
| <input checked="" type="checkbox"/> stredni_doba_zprac4 14.850.552 hodín | <input checked="" type="checkbox"/> stredni_doba_zprac9 6.624.578 hodín | <input checked="" type="checkbox"/> stredni_doba_zprac14 4.279.715 hodín | <input checked="" type="checkbox"/> stredni_doba_zprac19 3.165.779 hodín | <input checked="" type="checkbox"/> stredni_doba_zprac24 2.516.049 hodín | <input checked="" type="checkbox"/> stredni_doba_zprac29 2.109.765 hodín | <input checked="" type="checkbox"/> stredni_doba_zprac34 2.068.703 hodín | <input checked="" type="checkbox"/> stredni_doba_zprac39 2.066.51 hodín |
| <input checked="" type="checkbox"/> stredni_doba_cyklus5 139.151 hodín | <input checked="" type="checkbox"/> stredni_doba_cyklus10 69.888 hodín | <input checked="" type="checkbox"/> stredni_doba_cyklus15 46.784 hodín | <input checked="" type="checkbox"/> stredni_doba_cyklus20 35.179 hodín | <input checked="" type="checkbox"/> stredni_doba_cyklus25 28.251 hodín | <input checked="" type="checkbox"/> stredni_doba_cyklus30 24.428 hodín | <input checked="" type="checkbox"/> stredni_doba_cyklus35 24.216 hodín | <input checked="" type="checkbox"/> stredni_doba_cyklus40 24.202 hodín |
| <input checked="" type="checkbox"/> stredni_doba_zprac5 11.082.268 hodín | <input checked="" type="checkbox"/> stredni_doba_zprac10 5.960.157 hodín | <input checked="" type="checkbox"/> stredni_doba_zprac15 3.995.821 hodín | <input checked="" type="checkbox"/> stredni_doba_zprac20 3.065.168 hodín | <input checked="" type="checkbox"/> stredni_doba_zprac25 2.413.686 hodín | <input checked="" type="checkbox"/> stredni_doba_zprac30 2.087.313 hodín | <input checked="" type="checkbox"/> stredni_doba_zprac35 2.068.095 hodín | <input checked="" type="checkbox"/> stredni_doba_zprac40 2.066.181 hodín |

nastaveni_pocctu_desek
100

nastaveni_obaluha Dobsuha
false

Obr. 38, Experiment ParametersVariation